Transformer-based Multi-Stage Enhancement for Remote Sensing Image Super-Resolution

Sen Lei, Zhenwei Shi*, *Member IEEE*, Wenjing Mo

***Abstract*—Convolutional neural networks have made great breakthrough in recent remote sensing image super-resolution tasks. Most of these methods adopt upsampling layers at the end of the models to perform enlargement, which ignores feature extraction in the high-dimension space and thus limits super-resolution performance. To address this problem, we propose a new super-resolution framework for remote sensing image to enhance the high-dimensional feature representation after the up-sampling layers. We name the proposed method as Transformer-based Enhancement Network (TransENet), where transformers are introduced to exploit features at different levels. The core of the TransENet is a transformer-based multi-stage enhancement structure which can be combined with traditional super-resolution frameworks to fuse multi-scale high/low-dimension features. Specifically, in this structure, the encoders aim to embed the multi-level features in the feature extraction part and the decoders are used to fuse these encoded embeddings. Experimental results demonstrate that our proposed TransENet can improve super-resolved results and obtain superior performance over several state-of-the-art methods.**

***Index Terms*—Super-resolution, remote sensing images, deep convolutional neural networks, transformer**

I. Introduction

Image super-resolution (SR) is one kind of image processing technology, which aims to recover high-resolution (HR) images from low-resolution (LR) ones. It has been widely used in medical imaging [1], video monitoring [2] and remote sensing analysis [3, 4]. In the field of remote sensing, the ground targets in HR images own more clear edges and contours than the ones in LR images, and the HR images thus often play an important role in many high-level remote sensing tasks such as object detection [5], change detection [6] and semantic labeling [7]. Instead of developing physical imaging technologies, SR is an alternative way to effectively produce HR remote sensing images and has drawn much attention in recent years.

SR from one image is a typical ill-posed problem. Nowadays, most researchers leverage deep learning to obtain strong feature representations from a large amount of HR/LR image pairs [8]. Compared with the traditional learning-based algorithms such as neighborhood embedding-based [9], sparse representation-based [10, 11] and local linear regression-based [12, 13] methods, the deep learning-based methods can automatically learn powerful feature representations and produce HR images with clearer edges and contours. Many specific structures are further proposed to enhance the performance, such as the residual block [14], recurrent structure [15, 16], attention mechanism [17, 18].

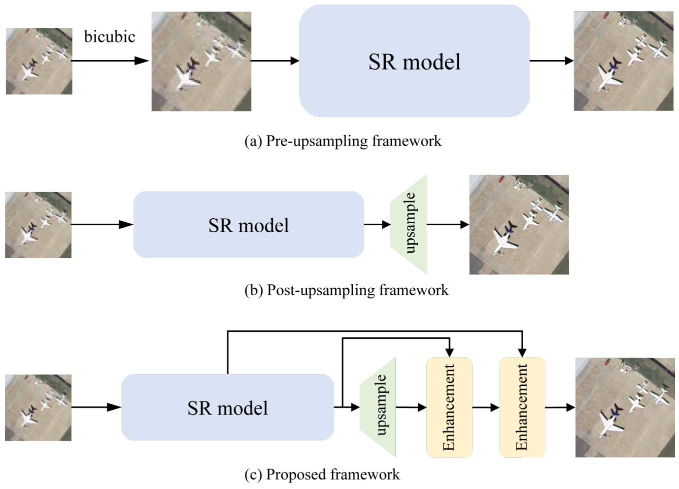
For deep learning-based SR methods, an up-sampling operation is utilized to enlarge the LR input. According to the position of the upsampling operation, these existing methods can be divided into two categories: pre-upsampling framework[19, 20] and post-upsampling framework[14, 21–24]. In this paper, we proposed a new SR framework for remote sensing images. All these frameworks are illustrated in Fig. 1.

The Pre-upsampling Framework. This framework is adopted widely at the early stages of deep learning-based SR algorithm. It first performs a interpolation operation (such as bicubic interpolation) on LR input and enlarges it to the same size as the HR reference. Then a SR model is used to recover the HR image from the interploated input. The SR model learns a nonlinear mapping between the interpolated LR input and the HR reference, without involving up-sampling operations, which reduces the learning difficulty to some extent. However, the computational cost significantly increases for a very deep network, since the feature extractions are all performed in the enlarged high-dimensional feature space.

The Post-upsampling Framework. In order to alleviate the problem of high computational cost, some researchers introduce the post-upsampling framework to construct an end-to-end SR architecture, in which the whole feature extractions are implemented in a low-dimensional space. For this purpose, the traditional up-sampling method is replaced with learnable upsampling layers, such as deconvolution [25] and sub-pixel convolution [26], which are inserted at the backend of the network and become one part of the SR model. Compared with the pre-upsampling framework, the computational cost reduction of this framework is proportional to the square of the preset magnification, where a feed-forward pass of a trained model can be significantly accelerated. This architecture design has become the mainstream in the image SR community [14, 21–24]. However, for this framework, the HR image will be directly recovered after up-sampling layers without further perform enhancement of feature expression. It increases the difficulty of training and restricts the improvement of recon-

The work was supported in part by the National Key Research and Development Program of China under the Grant 2019YFC1510905, in part by the National Natural Science Foundation of China under the Grant 62125102, and in part by the Beijing Natural Science Foundation under the Grant 4192034. (*Corresponding author: Zhenwei Shi*).

Sen Lei and Wenjing Mo are with AVIC Chengdu Aircraft Industrial (Group) Co. Ltd., Chengdu, China. Zhenwei Shi (Corresponding author) is with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China.



(a) Pre-upsampling framework

(b) Post-upsampling framework

(c) Proposed framework

Fig. 1. The illustration of different SR frameworks: (a) pre-upsampling framework (b) post-upsampling framework (c) the proposed framework.

struction accuracy especially for a large magnification.

Our Proposed Framework. To address the above problem, we propose a new SR framework for remote sensing images, named Transformer-based Enhancement Network (TransENet), which aims at making full use of high-dimensional and low-dimensional features to further enhance the feature representation ability after upsampling layers. Moreover, we introduce transformer model [27] to leverage the features of different stages. Compared with the traditional convolution, the transformer can capture long-distance dependencies and effectively mine the correlation between high/low-dimensional features. Meanwhile, in order to utilize multi-level information in remote sensing images, we further design a transformer-based multi-stage enhancement structure which consists of multiple encoders and decoders. Specifically, the encoders are used to encode features of different stages in the feature extraction part, and the decoders perform multi-stage fusion with high/low-dimensional features to strengthen the expressive ability of high-dimensional features. It should be noted that this structure can be combined with most SR methods based on the post-upsampling framework.

The main contributions of this paper are summarized as follows:

- We propose a new SR framework named TransENet for remote sensing images to enhance the high-dimensional feature representation after upsampling layers. Transformers are introduced to leverage features at different stages. Our design can further improve super-resolved results and obtain state-of-the-art SR performance on two public remote sensing dataset.
- We design a transformer-based multi-stage enhancement structure. This structure can be combined with traditional SR framework to fuse multi-scale high/low-dimension features, where encoders aim to embed the multi-level features in the feature extraction and decoders are used to fuse these encoded features. Comprehensive ablation experiments verify the effectiveness of this design.

The rest parts of this paper are organized as follows. In Section II, we provide detailed related works of image SR and transformer for image processing. The overview of the proposed TransENet and the Transformer-based multi-stage enhancement structure are carefully discussed in Section III. In Section IV, ablation studies and quantitative and qualitative results are presented. Finally, the conclusions are drawn in Section V.

II. Related Work

A. CNN-based Natural Image SR

In recent years, convolutional neural networks (CNN) have greatly boosted the development of the natural image SR community. Different from the traditional methods [10–13], CNN-based methods often attempt to build an end-to-end network to directly learn a linear mapping from the given LR input to the HR reference. The up-sampling operation is usually utilized to complete the enlargement of the input image. Based on the position of the upsampling operation in CNN models, these methods can be divided into two categories of the pre-upsampling framework based and the post-upsampling framework based. Early methods are most based on the pre-upsampling framework. SRCNN [8] is the first shallow convolutional neural network to recover high-frequency information from an upsampled LR image. Kim *et al.* [20] introduced a very deep convolutional network (VDSR) with 20 layers to learn the image residual between the HR reference and the upsampled LR one. Recent post-upsampling framework based methods often incorporate deconvolution layers or sub-pixel convolution layers into the SR network. FSRCNN [28] directly adopts the original LR image as input and uses a deconvolution layer at the end of the model to perform upsampling. Lim *et al.* [14] improved residual blocks by getting rid of batch normalization, and several residual blocks are stacked to construct feature extraction part followed by a upsample block with sub-pixel convolution layers. From then, many researchers denote to developing the feature extraction part to learn better representations on low-dimension feature space. Zhang *et al.* [17] introduced residual channel attention to exploit interdependencies among feature channels. Mei *et al.* [18] proposed a cross-scale non-local attention module to leverage the long-range feature-wise similarities.

B. SR for Remote Sensing Images

Nowadays remote sensing image SR has attracted much attention. In early time, sparse representation-based methods leaded the researches. Pan *et al.* [29] first introduced the sparse representation and combined structure self-similarity prior to perform remote sensing image SR. Hou *et al.* [30] proposed a global joint dictionary model to recover remote sensing HR images. Shao *et al.* [31] developed a coupled sparse autoencoder to better learn the mapping between LR images and HR ones with sparse representation coefficients. In recent years, the deep learning-based methods [22, 24, 31, 32] have achieved much better performance than these early sparse representation-based methods. LGCNet [32] is the first CNN-based model for remote sensing image SR, where local and global representations are both exploited to learn the image residual between HR images and the upscaled LR ones. Same with the trend in natural image SR field, most SR methods for remote sensing images adopt the post-sampling framework. Haut $et\ al.$ [22] combined residual units, skip connections and network-in-network structure to extract more informative features. Qin $et\ al.$ [23] introduced gradient maps to guide the proposed model to focus more on the edges of ground targets. Dong $et\ al.$ [31] proposed a second-order learning strategy to capture multi-scale feature information. Meanwhile, some works introduce attention mechanism to further improve reconstructed results. MSAN [33] extracts multi-level features via a multi-scale attention design, and a scene-adaptive SR strategy is adopted to make the MSAN to better handle different scenes. HSENet [24] exploits the hybrid-scale self-similarity information in the remote sensing images using non-local attentions. Moreover, many researchers introduced generative adversarial networks (GAN) to improve the visual quality of the super-resolved. Jiang $et\ al.$ [34] designed an edge-enhancement strategy to weaken the artifacts and noised caused by adversarial training. Lei $et\ al.$ [21] introduced coupled adversarial training to learn better discriminative ability and achieved better visual quality.

C. Transformer for Image Processing

Transformer [27] has been widely used in the filed of natural language processing [35–37] and more recently, many attempts have been made to get rid of convolutions and adopt transformer models into computer vision tasks. ViT [38] is a pure transformer-based image classification model and achieves the state-of-the-art. There are also some CNN-transformer hybrid works. DETR [39] combines CNN backbone and the encoder-decoder transformer to build a fully end-to-end detector without anchor generation and non-maximum suppression post-processing. SETR [40] treats semantic segmentation as a sequence-to-sequence prediction task where the transformer is leveraged to accomplish global context model. Meanwhile, some researchers also try to generalize the transformer to low-level visual tasks. Parmar $et\ al.$ [41] proposed Image Transformer to perform conditional image generation that can sequentially predict each pixel given its previous generated pixels. Jang $et\ al.$ [42] built the first GAN using purely transformers (TransGAN), free of any convolution operation, and it can achieve high quality image synthesis. Moreover, Chen $et\ al.$ [43] introduced a new pre-training model, namely, image processing transformer (IPT), to simultaneously handle many low-level computer vision tasks such as denoising, SR and deraining. IPT uses the encoder-decoder transformer as the main body of feature extraction part and is pre-trained on a large-scale dataset via contrastive learning. Different from IPT, our model aims to leverage the transformer to capture long range dependency between high-dimension and low-dimension features to enhance the final feature representation for remote sensing image SR.

III. Methodology

In this section, we introduce the Transformer-based Enhancement Network (TransENet) for remote sensing SR. The overall framework of TransENet is presented in Section III-A and the transformer-based multi-stage enhancement structure is carefully discussed in Section III-B. Besides, we will give a brief introduction to the implementation details in Section III-C.

A. Overview of TransENet

Fig. 2 illustrates the overall framework of our TransENet. Given a LR image I_{LR}, one convolutional layer is utilized to transform the input from RGB pixel space to feature space:

$$f_0 = Conv(I_{LR}) \tag{1}$$

where the $Conv$ denotes a convolutional operation and the f_0 represents initial feature which will be the input of the following low-dimensional feature extraction part.

As shown in Fig. 2, in the low-dimensional feature extraction part, we use several feature extraction modules (FEM) to extract high-frequency details of the ground targets in remote sensing images from different scales. Specifically, we consider two basic components including basic blocks and residual blocks. The structure of the FEM constructed by some basic blocks is shown in Fig. 3 (a). The basic block consists of a convolutional layer and a non-linear function ReLU and uses a local skip connection to ease the training special for a deep model. Moreover, Fig. 3 (b) shows the structure of the FEM constructed by some residual blocks. The residual block is borrowed from ResNet [44] and is widely used in the field of image SR reconstruction [14, 21]. In the experimental part, we will use these two kinds of structure to verify the effectiveness of the transformer-based multi-stage enhancement. The entire low-dimensional feature extraction part is defined as:

$$f_n = FEM_n(f_{n-1}) = FEM_n(FEM_{n-1}(...FEM_1(f_0)...)) \tag{2}$$

where FEM_n represents the n^{th} feature extraction module, and we use three FEMs in this paper considering of both speed and performance. Under this condition, the number of encoder modules is decided as 4 (3 for low-dimension feature embedding and 1 for high-dimension feature embedding) and the number of decoder modules is decided as 3.

After the feature extraction in the low-dimension feature space, we employ sub-pixel layer [26] to achieve the feature transformation from the low-dimension space to the high-dimension space.

$$f_{up} = Subpixel(f_n) \tag{3}$$

The low-dimension feature $f_1, ..., f_n$ and the high-dimension feature f_{up} will be the input of the proposed transformer-based multi-stage enhancement structure, where several encoders and decoders are applied to perform feature enhancement. It should be noted that we reduce the feature dimension via 1×1 convolution considering of the efficiency of the TransENet. Finally, one convolutional layer is applied to obtained the final super-resolved HR image I_{SR} based on the enhanced features.

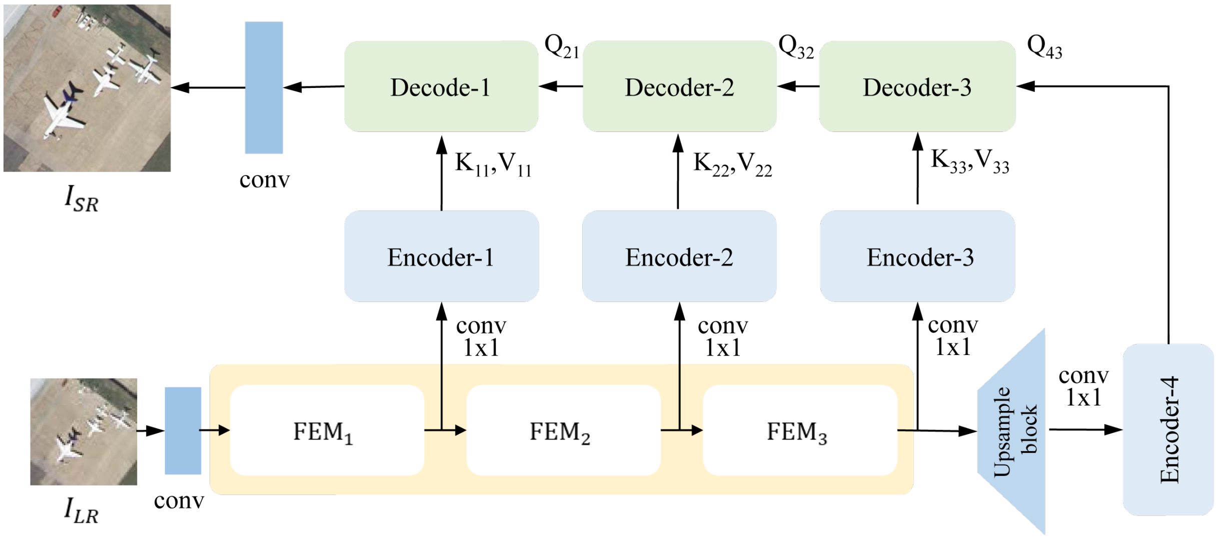


Fig. 2. The flowchart of the proposed method.

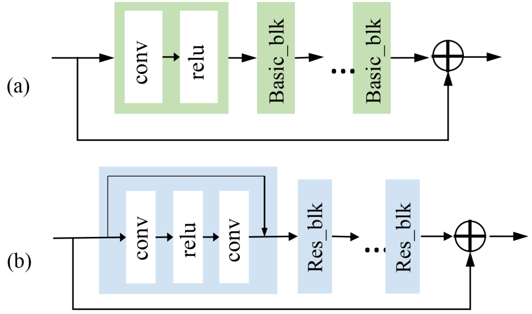


Fig. 3. The illustration of two kind of basic components including (a) the basic block and (b) the residual block.

We train the proposed model with L1 loss function. Given LR images I_{LR} and the corresponding HR reference I_{HR}, the loss function can be obtained as

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} ||I_{HR}^{(i)} - G_\theta(I_{LR}^{(i)})||_1. \tag{4}$$

where G_θ is the proposed model with parameters θ and $G_\theta(I_{LR}^{(i)})$ is exactly the aforementioned $I_{SR}^{(i)}$, and N is the number of training images.

B. Transformer-based Multi-stage Enhancement

In this subsection, we introduce a transformer-based multi-stage enhancement structure to enhance the representation ability of the high-dimension feature after upsampling layers. This structure can be combined with traditional SR frameworks to fuse multi-scale high/low-dimension features, which is shown in Fig. 2. We use several transformers consisted of encoders and decoders to capture long-distance dependencies and effectively mine the correlation between high/low-dimensional features. Here, we take Encoder-3 and Decoder-3 in the Fig. 2 as examples to provide a clear description about the process of the feature enhancement which are carefully illustrated in Fig. 4.

Transformer Encoder. The standard Transformer takes a set of 1D sequences of token embedding as input. In order to handle 3D features, we split the feature $f \in \mathbb{R}^{H \times W \times C}$ into some patches and reshape them into a sequence of vectors $f_{p_i} \in \mathbb{R}^{P_H P_W C}$, $i = \{1, ..., N\}$, where H, W, and C denote the height, the width and the number of channels of the feature maps, respectively. P_H and P_W are the height and the width of patches, and $N = \frac{HW}{P_H P_W}$ is the number of these patches and also is the length of the input sequence. Following [27, 38], the input vector size is usually fixed as D dimension, and we need to map f_{p_i} to D with a trainable linear projection. However, different from the setting in [27, 38], the positional embedding is not involved for each feature patches and more detailed discussions will be provided in the next experimental part. Thus the input of the transformer encoder can be represented as

$$y_0 = [f_{p_1}W, f_{p_2}W, ..., f_{p_2}W] \tag{5}$$

in which $W \in \mathbb{R}^{(P_H P_W C) \times D}$ is the linear projection matrix.

The main architecture of the encoder is following the original design in [27], which contains a multi-headed self-attention (MSA) module and a multi-layer perceptron (MLP) network. Referring to [38], we use the layer normalization (LN) [45] before each module and local residual structures are utilized. The architecture of Encoder-3 is carefully illustrated in Fig. 4 (a) and other encoders in our model have the same structure with the Encoder-3. The overall calculations of the encoder can be represented as

$$\begin{aligned} y_i^{'} &= MSA(LN(y_{i-1})) + y_{i-1}, i = 1, \ldots, L_e \\ y_i &= MLP(LN(y_i^{'})) + y_i^{'}, i = 1, \ldots, L_e \\ [f_{E_1}, f_{E_2}, &\ldots, f_{E_N}] = y_{L_e} \end{aligned} \tag{6}$$

where f_{E_i} is the output of the encoder corresponding to $f_{p_i}W$, which own the same dimension with $f_{p_i}W$. Besides, the MLP has two layers in which GELU [46] non-linear function is used.

These encoders can encode the features of different stages of the proposed model, and then some decoders are applied

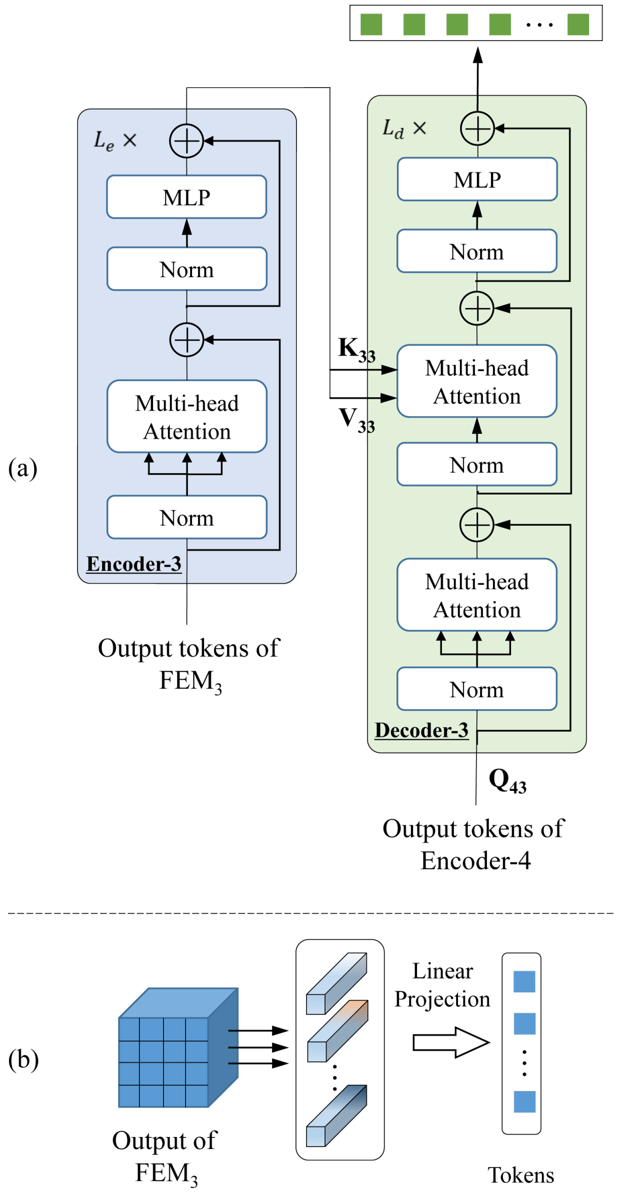


Fig. 4. The illustration of encoders and decoders including (a) the fusion stage with Encoder-3 and Decoder-3 and (b) the linear projection.

to fuse these embedding representations to enhance the high-dimensional feature.

Transformer Decoder. Comparing with the aforementioned encoder, apart from the MSA module and the MLP network, the transformer decoder also contains a specific MSA module with cross attention. This module can simultaneously handle the input features of the decoder and the output of the connective encoder, which is the core part of the decoder. The output of the decoder can be obtained as

$$
\begin{aligned}
& z_0 = [f_{E_1}, f_{E_2}, \ldots, f_{E_N}] \\
& z_i^{'} = MSA(LN(z_{i-1})) + z_{i-1}, i = 1, \ldots, L_d \\
& z_i^{''} = MSA(LN(z_{i-1}), LN(z_0)) + z_{i-1}^{'}, i = 1, \ldots, L_d \\
& z_i = MLP(LN(z_i^{''})) + z_i^{''}, i = 1, \ldots, L_d \\
& [f_{D_1}, f_{D_2}, \ldots, f_{D_N}] = y_{L_d}
\end{aligned}
\tag{7}
$$

where f_{D_i} is the output of the decoder, and L_d denotes the number of layers in the decoder.

Multi-Stage Enhancement. In order to make full use of the features extracted by the SR model in the low-dimensional space and combine the multi-scale information in the remote sensing image, we design a multi-stage feature enhancement, that is, multiple encoders are utilized to encode features at different levels. At the same time, multiple decoders are used to fuse and adjust the encoded features. The basic structure design is shown in Fig. 2.

Specifically, the feature extraction module $FEM_i(i = 1, 2, 3)$ extracts the feature representation after dimensionality reduction, and then enters the corresponding encoding module Encoder-i through block and linear mapping. The high-dimensional features after upsampling are encoded by the Encoder-4 module. In the subsequent feature enhancement process, high-dimensional features will be mainly used as the Q component in the Decoder, and the encoded low-dimensional features will be sequentially input as K and V into Decoder-1 to Decoder-3 to be combined with high-dimensional features. The combination process takes place in the multi-input MSA module in the Decoder can be formulated as

$$
\begin{aligned}
& Atten = softmax(QK^T/\sqrt{d_k}) \\
& MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \\
& wherehead_i = Atten(QW_i^Q, KW_i^K)VW_i^V
\end{aligned}
\tag{8}
$$

where d_k denotes the dimensions of features in these decoders, h is the heads of the MSA module and W_i^Q, W_i^K, W_i^V and W_i^O are all projection matrices. It should be noted that in Fig. 2 the subscript of Q/K/V variables is decided according to the index of the related components. Take Q_{43} for an example, the subscript of this variable is decided by the related Encoder-4 and Decoder-3.

C. Implementation Details

This paper focuses on remote sensing image SR at three magnifications of ×2, ×3 and ×4. In the training phase, 48 × 48 patches are randomly extracted from LR remote sensing images as well as the reference patches from their corresponding HR ones. Meanwhile, we use random rotation (90°, 180° and 270°) and horizontal flipping to augment the training samples. In the test phase, the LR test images are cropped into a set of 48 × 48 patches. We further use back-projection technology [47, 48] to reduce the blocking effect in the preliminary results, so as to obtain the final HR reconstructed images. The parameter settings of the encoder and the decoder in our model are listed in Table I. The number of layers in the encoder is set to 8, and that in the decoder is set to 1. The detailed analyses and experiments are provided in the next section.

For optimization, we use Adam optimizer [49] to train our model, where $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$. The initial learning rate is set to 10^{-4}, and mini-batch size is set to 16. The total training epochs is 2000 and the learning rate will decrease half at 1500. The proposed method is implemented

TABLE I
THE PARAMETER SETTINGS OF THE ENCODER AND THE DECODER IN OUR PROPOSED METHOD.

	Layers	Hidden size D	MLP dim	Heads	Head dim
Encoder	8	512	512	6	32
Decoder	1	512	512	6	32

by PyTorch[50], and all experiments are run on a NIVIDIA GeForce GTX 1080Ti graphics card. Our codes will be publicly available at https://github.com/Shaosifan/TransENet.

IV. EXPERIMENTAL RESULTS AND ANALYSES

A. Experimental Data set and Settings

In this paper, we use two public remote sensing data sets including UCMecred [51] and AID [52] to verify the effectiveness of the proposed method. These data sets have been widely used in the field of remote sensing SR [22, 23, 32].

- **UCMerced dataset** [51]. This dataset contains 21 classes of remote sensing scenes including agricultural, airplane, baseball-diamond, beach, and etc. There are 100 images for each class with a size of 256 × 256 pixels, and the spatial resolution of these images is 0.3 m/pixel. We split this data set into two halve for train and test, where 20% of the training set are taken as validation.
- **AID dataset** [52]. This dataset consists of 10000 image in 30 classes of remote sensing scenes including airport, bareland, church, dense-residential, and etc. All images are in 600 × 600 pixels, and the spatial resolution is up to 0.5 m/pixel. For AID data set, 80% of the whole dataset are randomly selected to be the training set, and the remaining images are used as the test set. Moreover, we randomly select 5 images per class in total of 150 images to construct the corresponding validation.

In our experiments, the original image in each data set is regarded as a real HR reference, and the corresponding LR image is obtained via the bicubic interpolation, so as to construct HR/LR image pairs for the training and evaluation. All results are measured by peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [53].

B. Ablation Studies

In this section, we conduct a series of experiments on the UCMereced dataset to explore the importance of each component in our method, where all models are trained with the same settings. For simplicity, these experiments are carried out with a magnification of ×4.

Effects of Encoders and Decoders. The encoders and decoders are key components of the proposed method. We investigate the effect of these components with the aforementioned basic blocks and residual blocks. Table II lists super-resolved results with different settings, where the number of layers of the encoders and decoders in our model is set to 1. Comparing with the baseline model, our method with encoders and decoders can achieve significant improvement both on basic blocks and residual blocks. Specifically, our method obtain 0.19 dB and 0.18 dB higher in term of PSNR than the baseline model with basic blocks and with residual blocks, respectively. It verifies the effectiveness and versatility of the proposed framework on different blocks. According to the results in Table II, residual blocks are finally used to construct the feature extraction modules, and the encoders and decoders are employed to enhance the features.

TABLE II
PSNR(dB) AND SSIM RESULTS WITH DIFFERENT COMPONENTS.

basic-blk	res-blk	De.	En.	PSNR	SSIM
✓	×	×	×	27.55	0.7547
✓	×	✓	×	27.75	0.7613
✓	×	✓	✓	27.74	0.7614
×	✓	×	×	27.59	0.7573
×	✓	✓	×	27.73	0.7614
×	✓	✓	✓	**27.76**	**0.7623**

TABLE III
PSNR(dB) AND SSIM RESULTS WITH DIFFERENT DECODER SETTINGS.

Decoder-3	Decoder-2	Decoder-1	PSNR	SSIM
×	×	×	27.59	0.7573
✓	×	×	27.72	0.7602
✓	✓	×	27.74	0.7616
✓	✓	✓	**27.76**	**0.7623**

Effects of Multi-stage Feature Enhancement. The design of multi-stage feature enhancement aims to leverage the multi-scale information in remote sensing images to obtain superior performance, where multiple decoders are involved to fuse high/low-dimensions feature stage-by-stage. Here, we investigate the effect of this design with different decoder configurations. It should be noted that when one certain decoder is added, the corresponding encoder will also be employed to fulfill feature embedding. Table III shows that the more decoders are involved, the better super-resolved performance will be achieved. At this time, more features at bottomed layers will passed to higher layers, and it relieves the difficulty in optimization and it beneficial to convergence of deep models. This phenomenon emphasizes the effectiveness of the multi-stage feature enhancement, and when these three decoders are used at the same time, the highest PSNR and SSIM values will be simultaneously obtained.

Is Positional Embedding Important for This Task? Position coding usually plays an important role in some transformer-based models, such as Bert [35], GPT [36] and ViT [38]. However, we find that the positional coding matters little in the proposed SR framework. To verify this point, we retrain the proposed model with or without learned positional encoding. In order to obtain a convincing comparison, we repeat the experiments for three times and report the mean and standard deviation of the results in Table IV, where P.E. denotes the position coding, En. and De. denote these encoders and decoders respectively. It can be observed that the position coding does not improve the reconstruction results, and the model with the P.E. trends to have a little lower PSNR and SSIM. We speculate that the reason for this phenomenon lies in the fact that the proposed model can implicitly learn the

TABLE IV
PSNR(dB) AND SSIM RESULTS WITH OR WITHOUT POSITION EMBEDDING.

P.E. in En.	P.E. in De.	PSNR	SSIM
×	×	**27.76** $\pm 8.05e^{-3}$	**0.7623** $\pm 4.12e^{-4}$
✓	×	27.73 $\pm 1.08e^{-2}$	0.7614 $\pm 1.70e^{-4}$
×	✓	27.72 $\pm 5.17e^{-3}$	0.7603 $\pm 2.52e^{-4}$
✓	✓	27.72 $\pm 8.02e^{-3}$	0.7603 $\pm 7.22e^{-4}$

TABLE V
RESULTS WITH DIFFERENT EN. AND DE. LAYERS SETTINGS

En. Layers	De. Layers	PSNR	SSIM
1	1	27.76	0.7623
1	2	27.73	0.7619
1	4	27.75	0.7626
1	8	27.72	0.7622
2	1	27.76	0.7624
4	1	27.75	0.7627
8	1	**27.77**	**0.7630**
12	1	27.75	0.7625

position information between different tokens to minimize the pixel-wise loss function in the training phase, and thus it is not necessary to add the learned positional encoding.

Number of Layers of Encoder and Decoder. The number settings of layers of the encoder and decoder can influence the final performance of our method. Therefore, we conduct a series of experiments about this point. Table V lists the reconstruction comparisons with different layer settings on the UCMerced test data set where the upscale factor is 4. We can see that when the numbers of layers of the encoder and the decoder are set to 8 and 1, TransENet can obtion the higher PSNR and SSIM. It implies that it is relatively harder to encode the low/high-dimension features than performing feature fusion.

Illustration of Training Processes. In order to present a different perspective about the transformer-based method, we illustrate the training processes of TransENet and the corresponding baseline model which is a pure CNN-based architecture. We plot the training curves in Fig. 5. Two interesting phenomena can be observed: the first is that at the initial training stage, the performance of the baseline model is better than TransENet, but our model would achieve better results after around 750 epochs; the second is that compared with the baseline model, the accuracy of TransENet is more volatile in the training process which is especially obvious at early stage. We guess that the reason for these two phenomena is that the CNN-based architecture has some inductive biases such as local receptive field and weight sharing, which makes it easier to learn for image processing tasks than the transformer-based method at the early stage. However, as the training progresses, the transformer-based method can gradually learn from the amount of image data and can obtain better recovered results.

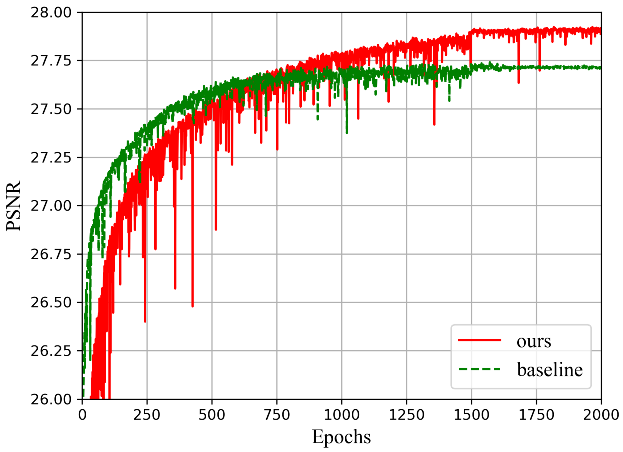


Fig. 5. The training curves of the proposed method and the baseline method on UCMerced dataset.

C. Comparisons with Other Methods

In this subsection, we compare the proposed method with some SR methods, including the classic bicubic interpolation, sparse coding (SC) [54], deep learning-based methods such as SRCNN [8], FSRCNN [28], VDSR [20], LGCNet [32], DCM [22] and DGANet-ISE [23]. Among them, SC, SRCNN, FSRCNN and VDSR are the approaches proposed for natural image SR task, while LGCNet, DCM and DGANet-ISE are recently proposed SR methods specifically designed for remote sensing images.

Quantitative Results on UCMerced Dataset. Table VI lists the results of these methods for upscale factor ×2, ×3 and ×4 on the UCMerced test dataset, where the best outcome is expressed in bold font. It should be noted that some results are reported in several published papers [22, 23]. It can be observed that TransENet obtain the highest value in term of PSNR and the second best in term of SSIM. Specifically, compared with other methods, the average PSNR value of our method at the three magnifications is 0.41 dB higher than DGANet-ISE, 0.44 dB higher than DCM, and 0.65 dB higher than LGCNet, 0.84 dB higher than FSRCNN, 1.15 dB higher than SRCNN, 1.39 dB higher than SC, and 2.62 dB higher than bicubic interpolation. For SSIM, the average performance of our method is 0.0039 lower than DGANet-ISE, 0.0048 higher than DCM, 0.0178 higher than LGCNet, 0.0236 higher than FSRCNN, 0.0310 higher than SRCNN, 0.0350 higher than SC, and 0.0731 higher than bicubic interpolation. The detailed results of different methods for the all 21 scene classes [1] of the UCMeced dataset is provided in Table VII at a upscale factor of 3. We can see that TransENet can achieved the best PSNR values in 12 scene classes, while DCM performed better in the other 9 categories. Compared with the DCM model, TransENet is more effective in some scenes with rich edges and contours, such as buildings, dense residential, storage tanks, and tennis court. At the same time, the overall PSNR of the method is

[1] All these 21 classes of UCMerced dataset: 1—Agricultural, 2—Airplane, 3—Baseballdiamond, 4—Beach, 5—Buildings, 6—Chaparral, 7—Denseresidential, 8—Forest, 9—Freeway, 10—Golfcourse, 11—Harbor, 12—Intersection, 13—Mediumresidential, 14—Mobilehomepark, 15—Overpass, 16—Parkinglot, 17—River, 18—Runway, 19—Sparseresidential, 20—Storagetanks, 21—Tenniscourt.

TABLE VI
MEAN PSNR (dB) AND SSIM OVER THE UCMERCED TEST DATA SET

scale	Bicubic	SC[54]	SRCNN[19]	FSRCNN[28]	LGCNet[32]	DCM[22]	DGANet-ISE[23]	Ours
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
2	30.76 / 0.8789	32.77 / 0.9166	32.84 / 0.9152	33.18 / 0.9196	33.48 / 0.9235	33.65 / 0.9274	33.68 / **0.9344**	**34.03** / 0.9301
3	27.46 / 0.7631	28.26 / 0.7971	28.66 / 0.8038	29.09 / 0.8167	29.28 / 0.8238	29.52 / 0.8394	– / –	**29.92** / **0.8408**
4	25.65 / 0.6725	26.51 / 0.7152	26.78 / 0.7219	26.93 / 0.7267	27.02 / 0.7333	27.22 / 0.7528	27.31 / **0.7665**	**27.77** / 0.7630

TABLE VII
MEAN PSNR (dB) OF EACH CLASS FOR UPSCALING FACTOR 3 ON UCMERCED TEST DATASET

class	Bicubic	SC [54]	SRCNN [19]	FSRCNN [28]	LGCNet [32]	DCM [22]	Ours
1	26.86	27.23	27.47	27.61	27.66	**29.06**	28.02
2	26.71	27.67	28.24	28.98	29.12	**30.77**	29.94
3	33.33	34.06	34.33	34.64	34.72	33.76	**35.04**
4	36.14	36.87	37.00	37.21	37.37	36.38	**37.53**
5	25.09	26.11	26.84	27.50	27.81	28.51	**28.81**
6	25.21	25.82	26.11	26.21	26.39	**26.81**	26.69
7	25.76	26.75	27.41	28.02	28.25	28.79	**29.11**
8	27.53	28.09	28.24	28.35	28.44	28.16	**28.59**
9	27.36	28.28	28.69	29.27	29.52	**30.45**	30.38
10	35.21	35.92	36.15	36.43	36.51	34.43	**36.68**
11	21.25	22.11	22.82	23.29	23.63	**26.55**	24.72
12	26.48	27.20	27.67	28.06	28.29	**29.28**	29.03
13	25.68	26.54	27.06	27.58	27.76	27.21	**28.47**
14	22.25	23.25	23.89	24.34	24.59	**26.05**	25.64
15	24.59	25.30	25.65	26.53	26.58	27.77	**27.83**
16	21.75	22.59	23.11	23.34	23.69	**24.95**	24.45
17	28.12	28.71	28.89	29.07	29.12	28.89	**29.25**
18	29.30	30.25	30.61	31.01	31.15	**32.53**	31.25
19	28.34	29.33	29.40	30.23	30.53	29.81	**31.57**
20	29.97	30.86	31.33	31.92	32.17	29.02	**32.71**
21	29.75	30.62	30.98	31.34	31.58	30.76	**32.51**
AVG	27.46	28.23	28.66	29.09	29.28	29.52	**29.92**

0.40 dB higher than that of DCM.

Quantitative Results on AID Dataset. In order to further verify the effectiveness of TransENet, we compare the proposed method with other methods on AID dataset. Different from the UCMerced dataset, this one is larger in amount and contains more scene categories in total of 30. The overall results of various methods on this dataset are shown in Table VIII. It can be seen that, compared with other methods, TransENet has the best results on these three magnifications. In addition, Table IX lists the detailed outcomes on the 30 classes [2] with the magnification of 4 and the average PSNR is measured. It shows that TransENet achieves the best results on all the ground target scenes. From Table VII and Table IX, it implies that when the size of data set increases, TransENet can obtain better results than DCM.

Qualitative Comparisons. In addition to quantitative comparison, we here provide a qualitative comparison of the recovered results with different methods. Fig. 6 shows some super-resolved examples of UCMerced dataset including 'airplane' and 'runway' scenes, and Fig. 7 presents some ones of AID dataset including 'stadium' and 'medium-residential' scenes. Overall, comparing with other methods, the proposed method can obtain better results with clearer edges and contours which are also closer to the HR references.

V. CONCLUSION

In this paper, we propose a new SR framework for remote sensing images, namely, Transformer-based Enhancement Network (TransENet). TransENet aims at making full use of high/low-dimensional features and enhance the high-dimensional feature representation after the upsampling layers. The core part of the TransENet is a transformer-based multi-stage enhancement structure which can be combined with traditional SR frameworks to fuse multi-scale high/low-dimension features. In our TransENet, encoders aim to embed the multi-level features in the feature extraction and decoders are used to fuse these encoded features. Ablation studies have verified the effectiveness of the multi-stage enhancement structure. Meanwhile, experimental results on two public data sets show that compared with some state-of-the-arts, our method can obtain better super-resolved results.

REFERENCES

[1] H. Greenspan, "Super-resolution in medical imaging," *The computer journal*, vol. 52, no. 1, pp. 43–63, 2009.

[2] L. Zhang, H. Zhang, H. Shen, and P. Li, "A super-resolution reconstruction algorithm for surveillance images," *Signal Processing*, vol. 90, no. 3, pp. 848–859, 2010.

[3] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[4] X. Xu, B. Pan, Z. Chen, Z. Shi, and T. Li, "Simultaneously multiobjective sparse unmixing and library pruning for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3383–3395, 2020.

[5] Z. Zou and Z. Shi, "Ship detection in spaceborne optical image with svd networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5832–5845, 2016.

[6] H. Chen, W. Li, and Z. Shi, "Adversarial instance augmentation for building change detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[2] All these 30 classes of AID dataset: 1—Airport, 2—Bareland, 3—Baseballdiamond, 4—Beach, 5—Bridge, 6—Center, 7—Church, 8—Commercial, 9—Denseresidential, 10—Desert, 11—Farmland, 12—Forest, 13—Industrial, 14—Meadow, 15—Mediumresidential, 16—Mountain, 17—Park, 18—Parking, 19—Playground, 20—Pond, 21—Port, 22—Railwaystation, 23—Resort, 24—River, 25—School, 26—Sparseresidential, 27—Square, 28—Stadium, 29—Storagetanks, 30—Viaduct.

TABLE VIII
MEAN PSNR (dB) AND SSIM OVER THE AID TEST DATA SET

scale	Bicubic	SRCNN[19]	LGCNet[32]	VDSR[20]	DCM[22]	Ours
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
2	32.39 / 0.8906	34.49 / 0.9286	34.80 / 0.9320	35.05 / 0.9346	35.21 / 0.9366	**35.28** / **0.9374**
3	29.08 / 0.7863	30.55 / 0.8372	30.73 / 0.8417	31.15 / 0.8522	31.31 / 0.8561	**31.45** / **0.8595**
4	27.30 / 0.7036	28.40 / 0.7561	28.61 / 0.7626	28.99 / 0.7753	29.17 / 0.7824	**29.38** / **0.7909**

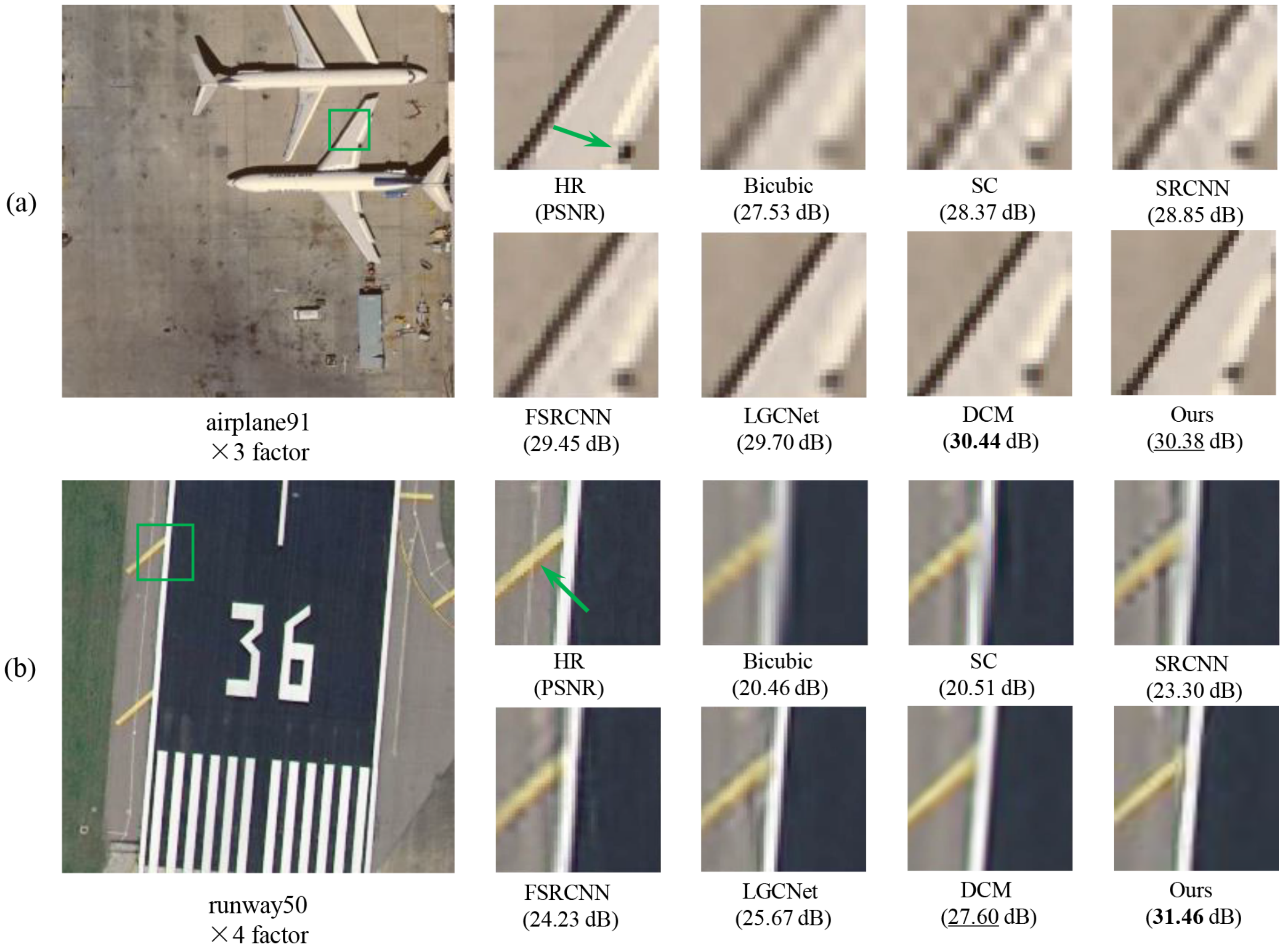


Fig. 6. Result comparisons on UCMerced dataset with different methods.

[7] S. Lei, Z. Shi, X. Wu, B. Pan, X. Xu, and H. Hao, "Simultaneous super-resolution and segmentation for remote sensing images," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 3121–3124.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[9] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[10] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[11] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE transactions on image processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[12] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1920–1927.

[13] ——, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian conference on com-*

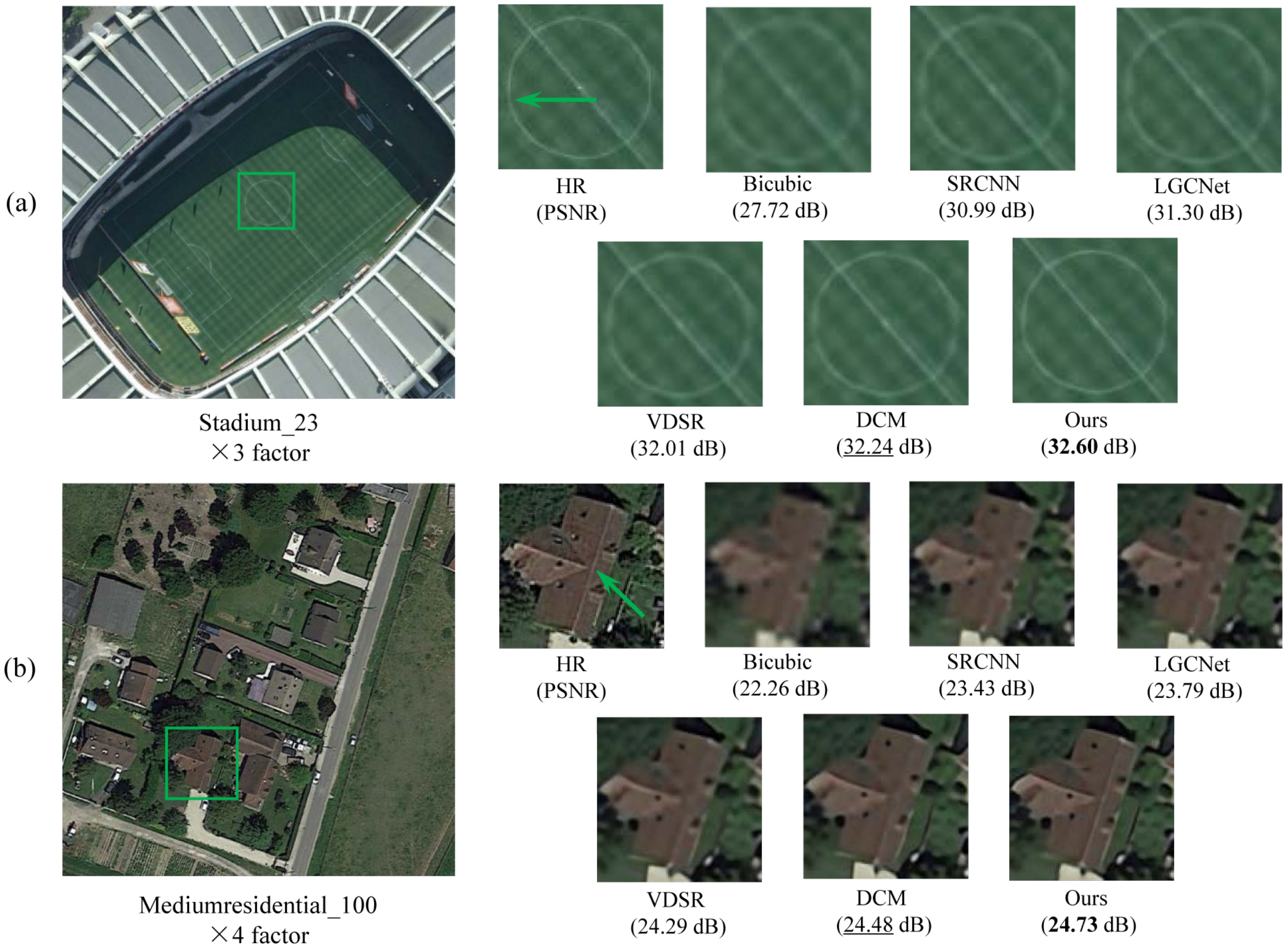


Fig. 7. Result comparisons on AID dataset with different methods.

puter vision. Springer, 2014, pp. 111–126.

[14] 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.

[15] 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.

[16] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4539–4547.

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

[18] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5690–5699.

[19] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[20] J. Kim, J. Kwon Lee, and K. Mu 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.

[21] S. Lei, Z. Shi, and Z. Zou, "Coupled adversarial training for remote sensing image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3633–3643, 2020.

[22] J. M. Haut, M. E. Paoletti, R. Fernández-Beltran, J. Plaza, A. Plaza, and J. Li, "Remote sensing single-image super-resolution based on a deep compendium model," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 9, pp. 1432–1436, 2019.

[23] M. Qin, S. Mavromatis, L. Hu, F. Zhang, R. Liu, J. Sequeira, and Z. Du, "Remote sensing single-image resolution improvement using a deep gradient-aware network with image-specific enhancement," *Remote Sens-*

TABLE IX
MEAN PSNR (dB) OF EACH CLASS FOR UPSCALING FACTOR 4 ON AID TEST DATASET

class	Bicubic	SRCNN [19]	LGCNet [32]	VDSR [20]	DCM [22]	Ours
1	27.03	28.17	28.39	28.82	28.99	**29.23**
2	34.88	35.63	35.78	35.98	36.17	**36.20**
3	29.06	30.51	30.75	31.18	31.36	**31.59**
4	31.07	31.92	32.08	32.29	32.45	**32.55**
5	28.98	30.41	30.67	31.19	31.39	**31.63**
6	25.26	26.59	26.92	27.48	27.72	**28.03**
7	22.15	23.41	23.68	24.12	24.29	**24.51**
8	25.83	27.05	27.24	27.62	27.78	**27.97**
9	23.05	24.13	24.33	24.70	24.87	**25.13**
10	38.49	38.84	39.06	39.13	39.27	**39.31**
11	32.30	33.48	33.77	34.20	34.42	**34.58**
12	27.39	28.15	28.20	28.36	28.47	**28.56**
13	24.75	26.00	26.24	26.72	26.92	**27.21**
14	32.06	32.57	32.65	32.77	32.88	**32.94**
15	26.09	27.37	27.63	28.06	28.25	**28.45**
16	28.04	28.90	28.97	29.11	29.18	**29.28**
17	26.23	27.25	27.37	27.69	27.82	**28.01**
18	22.33	24.01	24.40	25.21	25.74	**26.40**
19	27.27	28.72	29.04	29.62	29.92	**30.30**
20	28.94	29.85	30.00	30.26	30.39	**30.53**
21	24.69	25.82	26.02	26.43	26.62	**26.91**
22	26.31	27.55	27.76	28.19	28.38	**28.61**
23	25.98	27.12	27.32	27.71	27.88	**28.08**
24	29.61	30.48	30.60	30.82	30.91	**31.00**
25	24.91	26.13	26.34	26.78	26.94	**27.22**
26	25.41	26.16	26.27	26.46	26.53	**26.63**
27	26.75	28.13	28.39	28.91	29.13	**29.39**
28	24.81	26.10	26.37	26.88	27.10	**27.41**
29	24.18	25.27	25.48	25.86	26.00	**26.20**
30	25.86	27.03	27.26	27.74	27.93	**28.21**
AVG	27.30	28.40	28.61	28.99	29.17	**29.38**

ing, vol. 12, no. 5, p. 758, 2020.

[24] S. Lei and Z. Shi, "Hybrid-scale self-similarity exploitation for remote sensing image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[25] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[26] 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 subpixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[28] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*. Springer, 2016, pp. 391–407.

[29] Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma, and W. Sun, "Super-resolution based on compressive sensing and structural self-similarity for remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 9, pp. 4864–4876, 2013.

[30] B. Hou, K. Zhou, and L. Jiao, "Adaptive super-resolution for remote sensing images based on sparse representation with global joint dictionary model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2312–2327, 2017.

[31] 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 Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3473–3485, 2020.

[32] S. Lei, Z. Shi, and Z. Zou, "Super-resolution for remote sensing images via local-global combined network," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 8, pp. 1243–1247, 2017.

[33] S. Zhang, Q. Yuan, J. Li, J. Sun, and X. Zhang, "Scene-adaptive remote sensing image super-resolution using a multiscale attention network," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[34] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced gan for remote sensing image superresolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5799–5812, 2019.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[40] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6881–6890.

[41] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4055–4064.

[42] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two transformers can make one strong gan," *arXiv preprint arXiv:2102.07074*, 2021.

[43] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[45] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[46] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[47] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 349–356.

[48] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3118–3126.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[51] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.

[52] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[54] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.