Article

Remote Sensing Imagery Super Resolution Based on Adaptive Multi-Scale Feature Fusion Network

Xinying Wang ^{1,2}, Yingdan Wu ^{1,2,3,4,*}, Yang Ming ^{5,*} and Hui Lv ^{1,2,4}
¹ School of Science, Hubei University of Technology, No. 28 Nanli Road, Wuhan 430068, China; xiny_wang@163.com (X.W.); lvhui@hbut.edu.cn (H.L.)

² Hubei Collaborative Innovation Centre for High-Efficient Utilization of Solar Energy, Hubei University of Technology, No. 28 Nanli Road, Wuhan 430068, China

³ Center for Spatial Information Science and Systems, George Mason University, Fairfax, VA 22030, USA

⁴ Hubei Engineering Technology Research Center of Energy Photoelectric Device and System, Hubei University of Technology, No. 28 Nanli Road, Wuhan 430068, China

⁵ Institute of Surveying and Mapping, CCCC Second Highway Consultants Co., Ltd, No. 18 Chuangye Road, Wuhan 430056, China

* Correspondence: yd_wu2010@hbut.edu.cn (Y.W.); mingyang@ccshcc.cn (Y.M.); Tel.: +86-27-8453-3272 (Y.W.)

Received: 13 January 2020; Accepted: 17 February 2020; Published: 19 February 2020

Abstract: Due to increasingly complex factors of image degradation, inferring high-frequency details of remote sensing imagery is more difficult compared to ordinary digital photos. This paper proposes an adaptive multi-scale feature fusion network (AMFFN) for remote sensing image super-resolution. Firstly, the features are extracted from the original low-resolution image. Then several adaptive multi-scale feature extraction (AMFE) modules, the squeeze-and-excited and adaptive gating mechanisms are adopted for feature extraction and fusion. Finally, the sub-pixel convolution method is used to reconstruct the high-resolution image. Experiments are performed on three datasets, the key characteristics, such as the number of AMFEs and the gating connection way are studied, and super-resolution of remote sensing imagery of different scale factors are qualitatively and quantitatively analyzed. The results show that our method outperforms the classic methods, such as Super-Resolution Convolutional Neural Network (SRCNN), Efficient Sub-Pixel Convolutional Network (ESPCN), and multi-scale residual CNN (MSRN).

Keywords: super-resolution; remote sensing imagery; adaptive multi-scale feature fusion

1. Introduction

Image super-resolution (SR), is a classical yet challenging problem in the field of computer vision. The goal of image super-resolution is to reconstruct a visually pleasing high-resolution (HR) image from one or more low-resolution (LR) images [1]. Remote sensing imagery, captured from the satellite optical imaging sensors, provides abundant information to monitor the Earth's surface, having broad application in the fields of object matching and detection, land cover classification, assessment of urban economic levels, resource exploration, etc. It has proved that high-resolution remote sensing images play an important role. However, due to factors such as long-distance imaging, atmospheric turbulence, transmission noise, and motion blurring, the quality and the spatial resolution of remote sensing imagery are relatively poorer and lower as compared with natural images. Moreover, the ground objects of remote sensing imagery usually have different scales, causing the objects and surrounding environment to mutually couple in the joint distribution of their image patterns [2]. Therefore, super-resolution for remote sensing imagery has attracted huge interest and become a hot research topic.

The methods of image super-resolution can be divided into two groups: Multiple Image Super-Resolution (MISR) [3] and Single Image Super-Resolution (SISR) [4–6]. The former requires a set of LR images to reconstruct an HR image. SISR is more popular in practice, and the method proposed in this article belongs to the SISR.

To solve the problem of SISR, various algorithms, such as interpolation-based, reconstruction-based, and learning-based methods have been developed over the past few decades. In recent years, due to the rapid development of deep learning theory, the deep learning-based super-resolution methods have gradually become mainstream [7]. Super-Resolution Convolutional Neural Network (SRCNN) is the first deep learning algorithm for image super-resolution [8]. It treats each step of sparse coding as a convolutional neural network process, and directly establishes an end-to-end mapping between a pair of LR image patch and a HR one. However, it consumes much time for the input of the network a bicubic interpolation version of a LR image. Fast Super-Resolution Convolutional Neural Network (FSRCNN) [9] is the upgraded version with higher calculation efficiency, as it can input the original low-resolution image to the network directly by using deconvolution layer, and this makes the dimension of the input to the network relatively small, beneficial to reduce the computation and accelerate the reconstruction speed. In a similar way, Shi et al. [10] proposed Efficient Sub-Pixel Convolutional Network (ESPCN). Generally, the effective size of image context for reconstruction is correlated with the receptive field size of CNN, which depends on the convolutional kernel size in each layer and the depth of CNN. However, the above methods are essentially shallow networks, and they may have problems, such as small receptive field and insufficient image information for image reconstruction.

To solve this problem, Lin et al. proposed a dilated CNN to enlarge receptive field size for the image super-resolution [11]. Kim et al. [12] proposed Very Deep Super-resolution Convolutional Networks (VDSR) with 20 layers to obtain large receptive field, and learned only the residuals between the LR image and the HR image to accelerate the convergence speed. Tai et al. [13] proposed Deep Recursive Residual Network (DRRN), which arranged the basic residual units and blocks in a recursive topology to increase the depth of network and reduce parameter number. An increasing number of methods used the intermediate feature information of the network to improve the performance, such as DRCN [14], SRResNet [15], SRDenseNet [16], and MemNet [17]. Huang et al. [18] established a dense convolutional network (DenseNet), instead of passing the features of the previous layer to the next layer sequentially, and features extracted from all previous layers were passed to the current layer. Li et al. [19] proposed a multi-scale residual CNN (MSRN), which introduced multi-scale convolutional filters to enhance the feature inference capability. Xu et al. [20] proposed a global dense feature fusion convolutional network (DFFNet) for single image super-resolution of different scale factors, in which cascaded feature fusion blocks were used to learn global features in both spatial and channel direction. Deeper and more complicated networks are developed for image super-resolution.

Nowadays, many scholars apply CNN for the reconstruction of remote sensing imagery [21]. Lei et al. [22] proposed a multi-fork structure to extract local and global features of remote sensing images and obtain good reconstruction results. Xu et al. [23] used deep memory connection to combine image details of remote sensing images with environmental information. Jiang et al. [24] designed an ultra-dense residual network that uses the rich long- and short-line connections in the network to enhance the network's ability to extract remote sensing image features. Gu et al. [25] added a squeeze and excitation (SE) module to the network to improve the network representation. Dong et al. [26] used enhanced residual block and residual channel attention group to obtain multi-level remote sensing feature information. Lu et al. [2] reconstructed high-resolution remote sensing images by extracting patches of different sizes as multi-scale information input networks and fusing high-frequency information of different scales. However, the problem of redundant feature information is often ignored. In addition, the network structure for feature extraction and fusion is often fixed in above networks. Adaptive feature information extraction and fusion would be better for the remote sensing imagery super-resolution, due to complex factors of image degradation and diversity of image content.

For remote sensing image super-resolution, this paper proposes an Adaptive Multi-scale Feature Fusion Network (AMFFN). AMFFN can extract dense features directly from the original low-resolution image without any image interpolation preprocessing. Several adaptive multi-scale feature filtering blocks are cascaded to adaptively extract high-frequency detailed feature information of remote sensing imagery.

In summary, this paper contributes the following:

- (1) An adaptive multi-scale feature fusion network for the remote sensing image super-resolution, which can adaptively extract multi-scale feature information;
- (2) The mechanisms of squeeze-and-excited and adaptive gating are integrated for feature extraction and fusion, which can learn the channel correlation of feature maps, adaptively decide how much of the previous feature information should be reserved, reduce the redundant feature information among the intermediate multi-scale feature and enhance the use of useful feature information.

The remainder of this article is organized as follows. In Section 2, the network structure and the implementation details are discussed in detail. Section 3 demonstrates the experimental results on remote sensing image super-resolution, and the comparisons with other classical methods are discussed. The conclusions are given in Section 4.

2. Adaptive Multi-Scale Feature Fusion Network

2.1. Network Architecture

The network structure of AMFFN consists of four parts: Original feature extraction, adaptive multi-scale feature extraction, feature fusion and image reconstruction, as shown in Figure 1 and the part of adaptive multi-scale feature extraction is the core of our algorithm.

Figure 1. Network architecture of the proposed method.

The input of our network is the original low-resolution image for super-resolution, denoted as I_{LR} . A convolutional layer $conv$ with n_0 filters are firstly applied to the input image to produce a set of feature maps,

$$A_0 = w_0 * I_{LR} + b_0 \quad (1)$$

where A_0 is the original feature maps extracted from the low-resolution remote sensing imagery, w_0 corresponds filters in the convolutional layer, which is 128 filters with the spatial size of 3×3 in this paper, b_0 denotes the biases of the convolutional layer, and $*$ represents the convolution operation.

In the part of adaptive multi-scale feature extraction, supposing there are n adaptive multi-scale feature extraction (AMFE), and the output of i -th AMFE A_i can be represented as,

$$A_i = f_{MFE}(A_{i-1}) + g(A_{i-1}) \quad (1 \leq i \leq n) \quad (2)$$

where $f_{MFE}(\cdot)$ denotes the operation of multi-scale feature extraction, and $g(\cdot)$ represents adaptive feature gating operation, the details will be elaborated in the following sub-section. AMFE is the basic module for the adaptive feature extraction, which consists of a unit of multi-scale feature extraction (MFE) and a feature gating for adaptively retaining the feature information from the output of previous AMFE.

Through feature extraction, a series of feature maps, such as $A_0, \dots, A_n$, can be obtained. These feature maps contain a large amount of redundant information, which increase the computational burden significantly if they are directly used for image reconstruction. Therefore, before delivering these feature for super-resolution, a feature fusion layer is stacked after AMFE for feature fusion and reduction. The output of feature fusion layer A_{fusion} is formulated as,

$$A_{fusion} = w_f * [A_0, A_1, \dots, A_n] + b_f \quad (3)$$

where w_f corresponds to the weights of the feature fusion layer, which is 64 filters of a size of 1×1 , b_f is the corresponding biases, and $[A_0, A_1, \dots, A_n]$ denotes the concatenation of all feature maps extracted by the first feature extraction layer *conv* and AMFE.

As many CNN-based SISR methods, the sub-pixel convolution method is adopted to reconstruct the high-resolution image. The reconstruction function can be defined as follows,

$$I_{SR} = w_{s2} * shuffle(w_{s1} * A_{fusion}) \quad (4)$$

where w_{s1} denotes the weights of a 3×3 convolution layer. If the scale factor is r (e.g., $\times 2$), the number of filters in the convolution layer would be $C \cdot r^2$, and C refers to the channel number of the input feature maps. *shuffle*($\cdot$) represents the shuffling operation that rearranges the elements of a $H_{LR} \times W_{LR} \times C \cdot r^2$ tensor acquired in the top layer into a $rH_{LR} \times rW_{LR} \times C$ tensor, more details can consult to [10]. A 3×3 convolution layer w_{s2} with C_1 filters used to reconstruct the remote sensing images, and C_1 represents the number of channels of the original input image (e.g., if it is an RGB image, $C_1 = 3$). And the tensor of $rH_{LR} \times rW_{LR} \times C_1$ is our desired reconstructed high-resolution image I_{SR} . In our paper, L1 function is chosen to avoid introducing unnecessary training tricks and reduce computations.

2.2. Adaptive Multi-Scale Feature Extraction

As previously mentioned, the module of adaptive multi-scale feature extraction (AMFE) is the core module in our method. The structure of AMFE is illustrated as Figure 2, and it mainly consists of two units: Multi-scale feature extraction and filtering, and feature gating. For multi-scale unit feature extraction and filtering, it contains two parts: Multi-scale feature extraction and feature filtering.

Figure 2. The structure of AMFE module.

2.2.1. Multi-Scale Feature Extraction Unit

Firstly, the feature maps outputted from the previous AMFE are processed with a convolutional layer. For the i -th AMFE, this can be defined as,

$$M_i^0 = \phi(w_i^0 * A_{i-1} + b_i^0) \quad (5)$$

where A_{i-1} is the feature map from the previous AMFE, the number of feature maps outputted from each AMFE is 128 in the paper, w_i^0 corresponds to 128 filters of a size of $128 \times 3 \times 3$, b_i^0 is the corresponding biases, and $\phi(\cdot)$ represents activation function Relu.

Then, three types of filters $f_{i1} = 1 \times 1$, $f_{i2} = 3 \times 3$ and $f_{i3} = 5 \times 5$ is used to extract multi-scale features, the numbers of these filters are all 64, which can be expressed by Equation (6),

$$M_i^{1j} = \phi(w_i^{1j} * M_i^0 + b_i^{1j}) \quad (j = 1, 2, 3) \quad (6)$$

where j denotes the type index of the filters. Suppose that each filter bank contains $n_{i1} = n_{i2} = n_{i3} = 64$, and the convolutional output is concatenated and divided into 3 groups, that is $[M_i^0, M_i^{11}, M_i^{12}]$, $[M_i^0, M_i^{11}, M_i^{13}]$, $[M_i^0, M_i^{12}, M_i^{13}]$, as shown in Figure 2. Then, three different 1×1 convolution layers $\{w_i^{21}, w_i^{22}, w_i^{23}\}$ with 64 filters each are utilized to learn the channel correlation between the extracted multi-scale features of each group. The output feature maps $\{M_i^{21}, M_i^{22}, M_i^{23}\}$ are then concatenated to $[M_i^{21}, M_i^{22}, M_i^{23}]$, and a 1×1 convolution layer w_i^3 with 256 filters is used again to further extract the feature information of all. These process can be expressed by the following,

$$\begin{cases} M_i^{21} = w_i^{21} * [M_i^0, M_i^{11}, M_i^{12}] + b_i^{21} \\ M_i^{22} = w_i^{22} * [M_i^0, M_i^{11}, M_i^{13}] + b_i^{22} \\ M_i^{23} = w_i^{23} * [M_i^0, M_i^{12}, M_i^{13}] + b_i^{23} \end{cases} \quad (7)$$

$$M_i^3 = w_i^3 * [M_i^{21}, M_i^{22}, M_i^{23}] + b_i^3 \quad (8)$$

With filters of different spatial size and the cascaded structure of AMFE, we can build a hierarchical system that can extract multi-scale image feature information. The filter of spatial size 1×1 is mainly used to perform dimension reduction, but it also can learn the channel correlation between the feature maps, that is “extract” feature information along the channel direction.

2.2.2. Feature Filtering Unit

To enhance the sensitivity of informative features, feature filtering follows the multi-scale feature extraction. We borrowed the idea of squeeze-and-excitation (SE), proposed by the Hu et al. [27], to promote useful features and suppress less useful ones. The SE method firstly used global average pooling to generate channel-wise statistics, which was used as a channel descriptor. Then, two fully-connected (FC) layers around the non-linearity are used to form a bottleneck to derive the scalar corresponding each feature map. For high computation efficiency, the FC layers are replaced by the 1×1 convolution layer, and the diagram of feature filtering is illustrated in Figure 3. The operation of feature filtering unit can be defined as follows,

$$M_i^4 = w_i^4 * (A_{impor}(M_i^3) \times M_i^3) + b_i^4 \quad (9)$$

where $A_{impor}(\cdot)$ represents the operation of determining the importance score of each feature map, w_i^4 corresponds to 128 filters with spatial size of 1×1 .

Figure 3. Diagram of feature filtering.

2.2.3. Feature Gating Unit

When the structure of the network is fixed, it would be non-adaptation and not flexible enough to copy with the complex situation, especially for the remote sensing imagery. Therefore, in the module of adaptive multi-scale feature extraction, a simple feature gating mechanism is adopted in this paper, as illustrated by Figure 1. A shortcut connection enables the features outputted from the previous AMFE module to feed to current AMFE module directly. This is beneficial for reducing the loss of feature information during the transmission. In this paper, a feature gating mechanism is used to adaptively decide how much of the previous feature information should be reserved, and the implementation details are shown in the Figure 4.

Figure 4. Schematic diagram of feature gating unit.

The key for feature gating is how to adaptively obtain the value of gating score $score(A_{i-1})$ for the input feature A_{i-1} . When the value of gating score is determined, which is a scalar, then the reserved feature information A'_{i-1} is just as follows,

$$A'_{i-1} = g(A_{i-1}) = score(A_{i-1}) \times A_{i-1} \quad (10)$$

where $g(\cdot)$ represents the gating operation. To calculate the gating score and alleviate the calculation burden, the average pooling is used to reduce the dimension of the feature map, and use the global information to learn the gating score. Then, to capture the dependencies between channels, we add a simple non-linear function of two fully-connected layers connected with BatchNorm [28] and a ReLU activation function, and the output is a vector $\mathbf{V}$ of two elements. After softmax operation, vector $\mathbf{V}$ would be a normalized vector with $\mathbf{V}[0] + \mathbf{V}[1] = 1$. We define the second element $\mathbf{V}[1]$ is our desired value of gating score, which represents the how much proportion of feature information need to be reserved.

To enhance the robustness, the noise with Gumbel distribution is added when deriving the vector $\mathbf{V}$, that is the Gumbel-Softmax strategy [29] is used to replace the softmax. Then, the new vector $\mathbf{V}'$ is calculated as follows,

$$\mathbf{V}' = softmax((\mathbf{V} + \mathbf{G})/\tau) \quad (11)$$

where $\mathbf{G}$ is the Gumbel noise vector, each element G_i follows Gumbel(0,1) distribution, which can be sampled using inverse transform sampling by drawing $u_i \sim \text{Uniform}(0,1)$ and computing $G_i = -\log(-\log(u_i))$, τ is the softmax temperature, which is set to 1 in our paper.

3. Experimental Results and Analysis

3.1. Datasets and Performance Metrics

To verify the effectiveness of the method in this paper, three datasets of remote sensing imagery are used. The first is the UC Merced land-use dataset (referred to as the UC later) [30]. The dataset includes 21 types of scenes, and each scene has 100 images with size of 256×256 pixels and spatial resolution of 0.3 m. For each type of scene, 80 images are randomly selected into the training image set, and the remaining 20 images are selected into the test image set. The second is NWPU-RESISC45 (referred to as NW later) [28]. The dataset contains 45 types of scenes, each of which has 700 images with the same size of 256×256 pixels and the spatial resolution varying from 30 meters to 0.2 meters. For each type of scene, 100 images are randomly selected into the training image set, and 10 images are randomly selected from the remaining into the test image set. The third is the images captured by the satellite TianGong-2 (referred to as the TG later) [31]. The dataset consists of 6 types of scenes, and the total number of images is 2000, which are all selected into the test image set. Through these, we build our training image set and testing image set. The overall information and the example images of the experimental datasets are illustrated in Table 1 and Figure 5 respectively.

The algorithm is based on the PyTorch framework, which enables NVIDIA TitanXp Graphics Processing Unit (GPU) and Intel (R) Xeon (R) Silver 4116 Central Processing Unit (CPU) to train the model. The original high-resolution images are downsized by bicubic interpolation to generate corresponding low-resolution images for training, and the training images are augmented by horizontal or vertical flipping and 90° rotating transformation. For all training images, low-resolution patches with a size of 64×64 are extracted, and the total number of LR image patches is 11,124. In each training batch, we randomly extract 16 LR patches with the size of 64×64 and an epoch having 696 iteration of back-propagation. The maximum epoch number is 100, the learning rate is 0.0001, and the Adam optimizer is used. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are selected as the metrics for the evaluation of each experiment.

Table 1. Information of experimental data.

Dataset	Scene Number	Image Size (Pixels)	Image Number	Training Image Number	Testing Image Number	Spatial Resolution (m)
UC	21	256×256	2100	1680	420	0.3
NW	45	256×256	31,500	4500	450	0.2~30
TG	6	256×256	2000	0	2000	100

Figure 5. Example images of three datasets: (a) UC dataset, (b) NW dataset, (c) TG dataset.

3.2. Network Analysis

3.2.1. Number of AMFE Modules

AMFE module is the core part of our network, and its number dramatically affects the depth of our network. To find suitable value for it, the number of AMFE is set to 2, 4, 8, 16, 24, and 32, and the values of Loss and PSNR are depicted in Figure 6 and Table 2.

From Table 2, it can be found that the PSNR of reconstructed images increases with the number of AMFE modules. This can be explained by that with the increase of the number of AMFEs, more feature information could be extracted, which is beneficial for the super-resolution of remote sensing imagery. The increase rate is gradually slow down with the number of AMFEs. And when the number of AMFEs reaches to 16, our network reaches the best performance, as shown in Figure 6, the loss value decreases faster and is more stable than others, and the PSNR is the highest. When the number of AMFEs is 24 or 32, the network costs more time and has a relatively slow convergence, but a worse result. The overfitting and the insufficient training data may be the reason for this. For our experiment, we set the number of AMFEs to be 16 considering the tradeoff between the performance and computing efficiency.

Figure 6. Loss values of different numbers of AMFE with scale factor $\times 2$.

Table 2. PSNR and cost time of different numbers of AMFE with scale factor $\times 2$.

Number of AMFEs	2	4	8	16	24	32
PSNR(dB)	38.92	39.01	39.04	39.76	39.43	39.24
time(s)	0.12	0.14	0.17	0.24	0.31	0.37

3.2.2. Adaptive Feature Gating

To verify the effectiveness of our feature gating mechanism, three connection methods with the MFE are discussed, which are:

- 1) The output of MFE is directly used as the input of the next MFE;
- 2) Add a shortcut to connect the output of previous MFE with the input of the next MFE;
- 3) Replace the shortcut of the way 2) with our gating mechanism.

The comparison results are given in Figure 7 and Table 3. We can see that by adding the skip connection, it enables to directly learn the difference between the features and reaches a faster convergence speed, and with our gating mechanism, it shows better performance on both the convergence rate and the PSNR value. The convergence speed is faster than the short cut connection way after the 40 epochs and the PSNR is higher than other two methods by about 0.3 dB. The skip connection provides a shortcut to connect the output of previous MFE directly with the input of the next MFE, which is beneficial to the propagation of feature information, but it may result in information redundancy. In addition, excessive parameters might lead to overfitting. This maybe the reason that the skip connection method achieves worse result. Our feature gating strategy can learn from the practical images and adaptively determine the gating score, which decides how much proportion of feature information from previous MFE will be reserved and integrated. From the experimental results, we can find that the feature gating unit can reduce redundant information effectively and improve the performance of image super-resolution.

Figure 7. Loss values with different connection methods with scale factor $\times 2$.

Table 3. PSNR and cost time consumption with different connection methods with scale factor $\times 2$.

Connection Methods	1	2	3
PSNR(dB)	39.49	39.39	39.76
time(s)	0.20	0.20	0.24

3.3. Comparison Results with Other Classical Methods

Our proposed method AMFFN has been compared with classical methods, such as Bicubic interpolation, SRCNN [8], ESPCN [10], and MSRN [19]. The quantitative results of these methods for scale factor $\times 2$, $\times 3$, and $\times 4$ are in Table 4. To ensure fairness, SRCNN, ESPCN, MSRN and our network AMFFN are trained and tested by the same remote sensing image set.

Compared with SRCNN and ESPCN, the PSNR obtained by our method is higher by 3 dB to 5 dB, a significant improvement has been achieved. The reason for this is that our method can extract multi-scale feature and realize adaptive feature fusion, which contributes to the enhancement of results of image super-resolution. However, SRCNN and ESPCN are essentially shallow networks, with limiting ability of feature extraction and fusion. When contrasting to the MSRN method, which is also a deep network and achieves better results than SRCNN and ESPCN, our method outperforms it in terms of PSNR and SSIM.

Table 4. Comparison results with other classical methods.

Datasets	Scale Factor	Bicubic	SRCNN	ESPCN	MSRN	AMFFN
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
UC	$\times 2$	26.75/0.6402	29.67/0.7842	31.51/0.8184	34.83/0.9342	35.00/0.9360
	$\times 3$	24.32/0.4734	26.40/0.5842	27.70/0.6625	30.80/0.8539	30.94/0.8581
	$\times 4$	22.72/0.3402	24.72/0.4408	25.60/0.5221	28.61/0.7718	28.70/0.7772
NW	$\times 2$	27.94/0.6687	30.83/0.8156	32.46/0.8455	34.93/0.9296	35.30/0.9348
	$\times 3$	25.53/0.4977	27.71/0.6264	28.87/0.6967	31.32/0.8465	31.37/0.8477
	$\times 4$	23.95/0.3599	26.08/0.4804	26.85/0.5580	29.42/0.7746	29.47/0.7763
TG	$\times 2$	32.13/0.7219	35.48/0.8629	37.22/0.8477	40.40/0.9678	40.55/0.9682
	$\times 3$	29.32/0.5617	31.49/0.6974	33.10/0.7357	35.79/0.9062	35.84/0.9067
	$\times 4$	27.50/0.4185	29.52/0.5492	30.70/0.6115	33.34/0.8413	33.36/0.8420

Visual comparisons on scale factor $\times 2$ are shown in Figures 8–12. From the results, it is found that AMFFN can clearly reconstruct the green plants and the straight strips in the farm field in UC dataset. However, the methods of Bicubic interpolation, SRCNN and ESPCN cannot accurately reconstruct the green plants. MSRN can reconstruct some green plants, but the ringing effects arises when reconstructing the straight strips in the farmland. For the images of urban scene in the NW dataset and mountain and farmland scenes in the TG datasets, the linear features and spatial structure of reconstructed high-resolution images are clearer using our method.

Figure 8. Results of farmland scenes in UC dataset of different methods with scale factor $\times 2$.

Figure 9. Cont.

Figure 9. Results of road scenes in UC dataset of different methods with scale factor $\times 2$.

Figure 10. Results of building scenes in NW dataset of different methods with scale factor $\times 2$.

Figure 11. Cont.

Figure 11. Results of mountain scenes in TG dataset of different methods with scale factor $\times 2$.

Figure 12. Results of farmland in TG dataset of different methods with scale factor $\times 2$.

4. Conclusions

This paper proposes an adaptive multi-scale feature fusion network for remote sensing imagery. Several adaptive multi-scale feature extraction (AMFFN) are used to extract multi-scale feature information, and the squeeze-and-excited and feature gating unit mechanism are adopted to enhance the adaptation of feature information, to adaptively select and make full use of intermediate feature information. Quantitative and visual benchmarking results on different test data sets show that our AMFFN outperform the classical image super-resolution methods.

Author Contributions: Y.W. reviewed and edited the original draft. X.W. conceptualized the whole structure of the idea, developed the algorithm, and crafted the manuscript. Y.M. supervised the experiment and conducted the primary data analysis. H.L. performed the experimental results analysis and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61301278), and by Natural Science Foundation of Hubei Province (2018CFB540), and by Philosophy and Social Science Foundation of Hubei Province (19Q062), and by Open Foundation of Hubei Collaborative Innovation Centre for High-efficient Utilization of Solar Energy (HBSKFM2014001), and by China Scholarship Council (No. 201808420417).

Acknowledgments: The research is supported by the Center for Spatial Information Science and Systems of George Mason University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, C.; Ma, C.; Yang, M. Single-Image Super-Resolution: A benchmark. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 372–386.
2. Lu, T.; Wang, J.; Zhang, Y.; Wang, Z.; Jiang, J. Satellite image super-resolution via multi-scale residual deep neural network. *Remote Sens.* **2019**, *11*, 1588. [\[CrossRef\]](#)
3. Xu, J.; Liang, Y.; Liu, J.; Huang, Z. Multi-frame super-resolution of Gaofen-4 remote sensing images. *Sensors* **2017**, *17*, 2142.
4. Du, X.; Qu, X.; He, Y.; Guo, D. Single image super-resolution based on multi-scale competitive convolutional neural network. *Sensors* **2018**, *18*, 789. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Timofte, R.; Rothe, R.; Gool, L.V. Seven Ways to Improve Example-Based Single Image Super Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1865–1873.
6. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
7. Yang, W.; Zhang, X.; Tian, Y.; Wang, W.; Xue, J. Deep learning for single image super-resolution: A brief review. *IEEE Trans. Multimed.* **2019**, *21*, 3106–3122. [\[CrossRef\]](#)
8. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a Deep Convolutional Network for Image Super-Resolution. In Proceedings of the ECCV 2014, Zurich, Switzerland, 6–12 September 2014.
9. Dong, C.; Loy, C.C.; Tang, X. Accelerating the Super-Resolution Convolutional Neural Network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 391–407.
10. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016.
11. Lin, G.; Wu, Q.; Qiu, L.; Huang, X. Image super-resolution using a dilated convolutional neural network. *Neurocomputing* **2018**, *275*, 1219–1230. [\[CrossRef\]](#)
12. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016.
13. Tai, Y.; Yang, J.; Liu, X. Image Super-Resolution via Deep Recursive Residual Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3147–3155.
14. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-Recursive Convolutional Network for Image Super Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
15. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv* **2016**, arXiv:1609.04802.
16. Tong, T.; Li, G.; Liu, X.; Gao, Q. Image Super-Resolution Using Dense Skip Connections. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
17. Tai, Y.; Yang, J.; Liu, X.; Xu, C. MemNet: A Persistent Memory Network for Image Restoration. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4549–4557.
18. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
19. Li, J.C.; Fang, F.M.; Mei, K.F.; Zhang, G.X. Multi-Scale Residual Network for Image Super-Resolution. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.

20. Xu, W.; Chen, R.; Huang, B.; Zhang, X.; Liu, C. Single image super-resolution based on global dense feature fusion convolutional network. *Sensors* **2019**, *19*, 316. [CrossRef]
21. Mario, H.J.; Ruben, F.B.; Paoletti, M.E.; Javier, P.; Antonio, P.; Filiberto, P. A new deep generative network for unsupervised remote sensing single-image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6792–6796.
22. Lei, S.; Shi, Z.; Zou, Z. Super-resolution for remote sensing images via local–global combined network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1243–1247. [CrossRef]
23. Xu, W.; Xu, G.; Wang, Y.; Sun, X.; Lin, D.; Wu, Y. High Quality Remote Sensing Image Super-Resolution Using Deep Memory Connected Network. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 8889–8892.
24. Jiang, K.; Wang, Z.; Yi, P.; Jiang, J.; Xiao, J.; Yao, Y. Deep distillation recursive network for remote sensing imagery super-resolution. *Remote Sens.* **2018**, *10*, 1700. [CrossRef]
25. Gu, J.; Sun, X.; Zhang, Y.; Fu, K.; Wang, L. Deep residual squeeze and excitation network for remote sensing image super-resolution. *Remote Sens.* **2019**, *11*, 1817. [CrossRef]
26. Dong, X.; Xi, Z.; Sun, X.; Gao, L. Transferred multi-perception attention networks for remote sensing image super-resolution. *Remote Sens.* **2019**, *11*, 2857. [CrossRef]
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *arXiv* **2018**, arXiv:1709.01507.
28. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]
29. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv* **2016**, arXiv:1611.01144.
30. Yang, Y.; Newsam, S. 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, San Jose, CA, USA, 3–5 November 2010; pp. 270–279.
31. Manned Spaceflight Application Data Promotion Service Platform. Available online: <http://www.msadc.cn/> (accessed on 7 December 2019).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).