

# Nonpairwise-Trained Cycle Convolutional Neural Network for Single Remote Sensing Image Super-Resolution

Haopeng Zhang<sup>1</sup>, Member, IEEE, Pengrui Wang, Member, IEEE, and Zhiguo Jiang<sup>1</sup>, Member, IEEE

**Abstract**—Single image super-resolution (SISR) is to recover the high spatial resolution image from a single low spatial resolution one, which is a useful procedure for many remote sensing applications. Most previous convolutional neural network (CNN)-based methods adopt supervised learning. However, paired high-resolution and low-resolution remote sensing images are actually hard to acquire for supervised learning SR methods. To handle this problem, we propose a novel cycle convolutional neural network (Cycle-CNN). Our network consists of two generative CNNs for down-sampling and SR separately and can be trained with unpaired data. We perform comprehensive experiments on panchromatic and multispectral images of the GaoFen-2 satellite and the UC Merced land use data set. Experimental results indicate that our method achieves state-of-the-art CNN-based SR results and is robust against noise and blur in remote sensing images. Comprehensively considering super-resolved image quality and time costs, our proposed method outperforms the compared learning-based SISR approaches.

**Index Terms**—Convolutional neural network (CNN), nonpairwise training, remote sensing image, super-resolution (SR).

## I. INTRODUCTION

THE spatial resolution of imaging sensors aboard earth-observing satellites has been constantly improved [1] in recent years; nevertheless, the resolution still cannot satisfy the demand under certain circumstances. Physically increasing the spatial resolution may reduce the incoming light and the signal-to-noise ratio of the sensor, which causes the quality of the final image to decrease sharply. Furthermore, the cost of imaging sensors will increase greatly with the reduction of physical pixels or the enlargement of the instantaneous field of view (IFoV). Therefore, algorithmic-based methods are

more appropriate because they can improve image resolution beyond the limits of imaging sensors [2]. Super-resolution (SR) is a classical problem in image processing to obtain high-resolution (HR) images from low-resolution (LR) ones. The purpose of SR is to improve the spatial resolution of the LR images. Compared to LR images, the reconstructed images can provide more graphic details, and have better visual quality. Other than improving the perceptual quality, SR also contributes to improving other computer vision tasks, such as object detection [3], target recognition [4], and image segmentation [5], [6]. Hence, SR is widely applied in remote sensing field and promotes the development of the applications of remote sensing [7].

According to the numbers of input images, SR methods can be classified into two kinds, i.e., single image super-resolution (SISR) [8] and multiple images super-resolution (MISR) [9], [10]. Compared to MISR, SISR is more widely used in remote sensing, because SISR is appropriate for most kinds of imaging sensors beyond multiple images of the same scene [11]. In this research, we focus on SISR for remote sensing.

SISR is a classical ill-posed problem, since the solution is not certain and unique. Previous SR methods are mainly based on interpolation and reconstruction. Nearest, bilinear, bicubic [12], and lanczos3 resampling [13] are common interpolation methods. Methods based on reconstruction mainly include iterative back-projection (IBP) [14], maximum a posterior (MAP) [15], and projection onto convex sets (POCS) [16]. These methods constrain the reconstructed image through a priori model. Currently, learning-based SR methods have been proposed, such as methods based on sparse coding [17], [18] and anchored neighborhood regression (ANR) [19]. Especially, with the development of convolutional neural networks (CNNs), deep learning-based SR methods have become more and more popular, e.g., SRCNN [20], VDSR [21], and SRResNet [22]. It has been shown that deep learning-based methods have a powerful ability to reconstruct LR images. Most of the deep learning-based SR methods are trained by LR–HR pairs, i.e., LR and HR images from the same area. However, in practice, real LR–HR pairs are hard to acquire, because a remote sensing sensor usually does not simultaneously capture LR and HR images. Therefore, supervised methods usually use down-sampled images from HR as LR to train the network, e.g., bicubic down-sampling. Since real LR images

Manuscript received November 24, 2019; revised March 23, 2020, May 11, 2020, and June 9, 2020; accepted June 28, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0501300, Grant 2016YFB0501302, and Grant 2019YFC1510905, in part by the National Natural Science Foundation of China under Grant 61501009, and in part by the Fundamental Research Funds for the Central Universities. (Corresponding author: Haopeng Zhang.)

The authors are with the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University, Beijing 102206, China, also with the Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education, 102206 Beijing, China, and also with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 102206, China (e-mail: zhanghaopeng@buaa.edu.cn; wprui@buaa.edu.cn; jiangzg@buaa.edu.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2020.3009224

may not be generated from HR by the specific down-sampling way, the well-trained supervised network may not work well when reconstructing real remote sensing images. Therefore, creating a nonpairwise-trained network is essential and practically valuable for remote sensing image SR. In this article, we propose a nonpairwise-trained network named Cycle-CNN for remote sensing image SR. Our method can be trained with unpaired remote sensing images. The whole pipeline consists of two cycle modules. The first module is used to map LR images to HR images, i.e., SR, while the second module maps HR images back to LR images, like down-sampling. We performed experiments using panchromatic and multispectral images from the GaoFen-2 satellite. Experimental results show that our method outperforms other state-of-the-art supervised methods for SR of real remote sensing images. It should be noticed that this article is an extended version of our contribution [23] previously represented in the 2019 IEEE International Geoscience and Remote Sensing Symposium at Yokohama, Japan.

The organization of the rest of this article is as follows. Section II presents supervised SR methods and their limitations in remote sensing, as well as a brief introduction to unsupervised SR methods. Section III describes details of our Cycle-CNN. Section IV shows experimental results to validate the effectiveness and robustness. Conclusion is provided in Section V.

## II. RELATED WORKS

### A. Supervised SR Methods and Their Limitations in Remote Sensing

Most of the supervised learning-based SR methods aim to solve the following problem [24]:

$$\hat{z} = \arg \min_z \|z \downarrow_s - x\| + \lambda \Phi(z) \quad (1)$$

where  $x$  represents LR and  $z$  represents HR,  $s$  is down-sampling model, and  $\Phi(z)$  is the regularization term, and  $\lambda$  is the trade-off parameter, most of supervised methods use bicubic as a down-sampling model.

However, real LR remote sensing images are different from natural images. Because the satellite is far from target and the imaging system is under the influence of the atmosphere and shake, the remote sensing images suffer from more blur and noise than natural images and the degradation model is more complex and hard to estimate. Furthermore, real LR–HR pairs are hard to acquire. These lead training LR–HR pairs used in learning-based SR approaches have quite deviation from the real ones. As a result, the reconstructed images may not meet expectations [25]. Therefore, the traditional supervised learning-based SR methods are limited in the application of reconstructing real remote sensing images [26].

### B. Unsupervised SR Methods

In order to solve the shortcomings of supervised-based SR methods, unsupervised SR reconstruction methods have become a research hot spot in the field of SR. Previous unsupervised methods are mainly based on reconstruction

(RE), e.g., IBP [14], MAP [15], [27], Gaussian process regression (GPR) [28]. Methods based on reconstruction constrain the reconstructed image through the prior image model and the reconstruction process. The principle of IBP is based on the inverse projection of the analog error, minimizing the error of the super-resolved image through the degraded model and the LR image. MAP applies Bayesian theory, using prior knowledge in the form of prior probability functions to solve the problem of SR. In the GPR method, each pixel is predicted by its neighbors through the GPR. These methods do not rely on training data and are robust to noise and blur. However, the methods have limited ability to reconstruct high-frequency details of the images [29].

In recent years, some studies focused on image self-similarity-based unsupervised SR algorithms in which the training process of the methods only uses LR images. The fundamental of the methods is that images have strong internal data repetition [30]. The methods try to search and extract repetitive structures within the same scale and over different scales, and train the reconstruction process by the extracted LR/HR patches [31]. There are different ways to search the patches and reconstruct the image. Huang *et al.* [32] proposed SR from transformed self-exemplars (SelfEx). Their method can extract image patches sufficiently by perspective distortion and additional affine transformation. Shocher *et al.* [33] introduced zero-shot SR (ZSSR). It evaluates the kernel directly [34] from the test image, trains an image-specific CNN to reconstruct the test image LR from its lower-resolution version, and then applies the trained CNN to reconstruct the desired HR output. Haut *et al.* [35] proposed a deep generative network for unsupervised remote sensing image SR, which used a 2-D-CNN architecture model to generate HR image, and updated HR image iteratively by minimizing the mean square error (MSE) loss between the down-sampled images of the generated HR and real LR. Methods based on image self-similarity have better reconstruction quality than RE; however, the methods require a longer time because they need to train a single network for each image during testing. Furthermore, the SR results highly depend on the input image, so the methods cannot perform well when the input image does not have enough useful LR/HR connections. Because HR images are not used for training, the details of the image are still not well reconstructed.

To solve the disadvantages of the abovementioned methods, recently, researchers paid more and more attention to unsupervised learning-based SR methods using unpaired LR–HR images. Generative adversarial net (GAN) [36] is an effective model to solve the problem. Zhu *et al.* [37] proposed CycleGAN, which made sense in solving unpaired image-to-image translation problem. Inspired by CycleGAN, Yuan *et al.* [38] proposed CinCGAN, using two CycleGANs to reconstruct the images. The first one is noisy LR  $\leftrightarrow$  clean LR, and the second is clean LR  $\leftrightarrow$  clean HR. Bulat *et al.* [39] trained the HR-to-LR degradation model by unpaired LR–HR images and used the degradation model to generate LR images, and then used the generated images and HR images to train the SR network. In this work, we propose a learning-based SISR model trained by unpaired LR–HR remote sensing images.

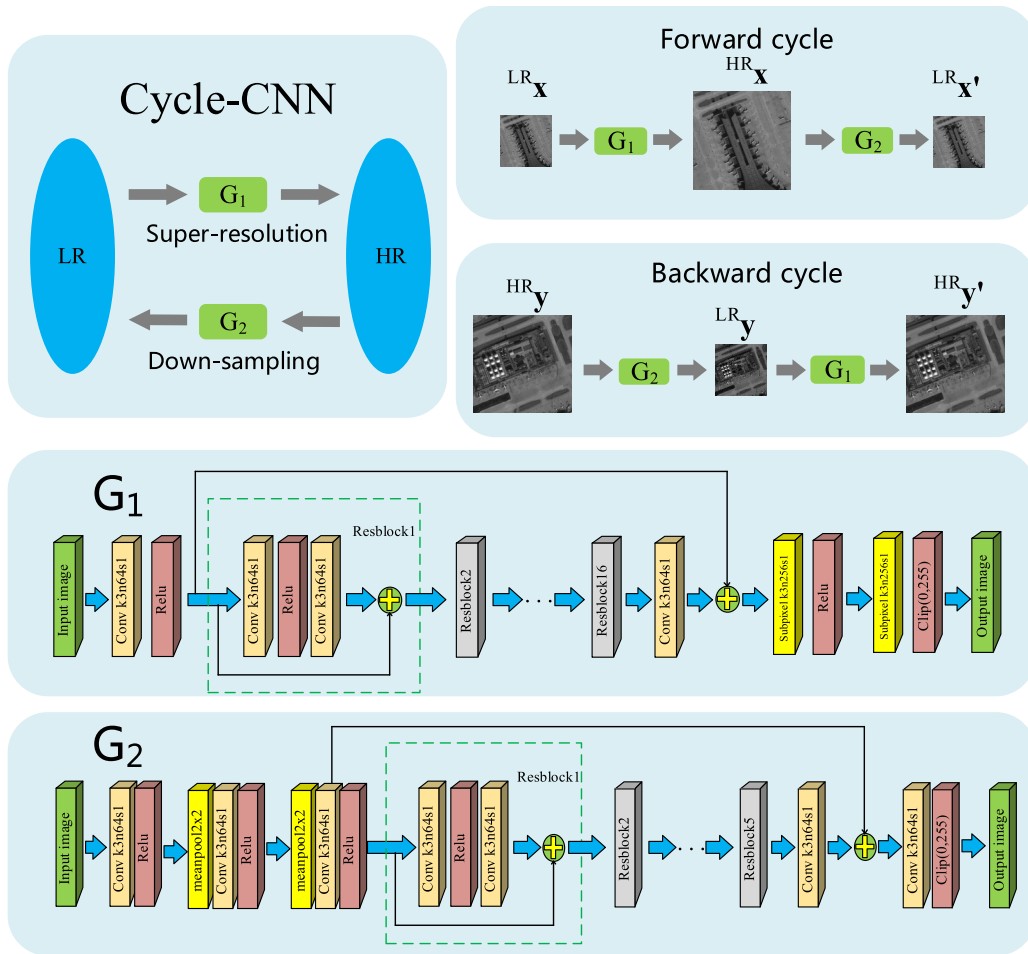


Fig. 1. Architecture of our Cycle-CNN network. Our network consists of two generative CNNs for SR ( $G_1$ ) and down-sampling ( $G_2$ ) separately.  $LR_x$  and  $HR_y$  are unpaired training data,  $HR_x$  and  $LR_y$  are their corresponding SR or down-sampling results, and  $LR_{x'}$  and  $HR_{y'}$  are the intermediate results of  $G_2$  and  $G_1$ . The numbers after  $k$ ,  $n$ , and  $s$  in the convolution blocks represent the kernel size, the number of filters, and the stride size, respectively.

### III. METHOD

#### A. Framework

The framework of our Cycle-CNN network is shown in Fig. 1. There are two generative CNNs in our network: one is used for image SR, and the other is used for image down-sampling. The whole network consists of two generators  $G_1$  and  $G_2$ . Starting with LR input, the CNN  $G_1$  for SR restores the input LR image  $LR_x$  to a HR image  $HR_x$ , and then the CNN  $G_2$  for down-sampling converts the HR image  $HR_x$  back to a LR image  $LR_{x'}$ .  $LR_x \rightarrow HR_x \rightarrow LR_{x'}$  is the forward cycle. Starting with the HR input,  $G_2$  down-samples the input HR image  $HR_y$  to a LR image  $LR_y$ , and then  $G_1$  converts the LR image  $LR_y$  back to a HR image  $HR_{y'}$ .  $HR_y \rightarrow LR_y \rightarrow HR_{y'}$  is the backward cycle. The detailed architectures of  $G_1$  and  $G_2$  are shown in the bottom half of Fig. 1. It should be noticed that the input HR image  $HR_y$  in unpaired training is not the corresponding ground truth HR of the input LR image  $LR_x$ . There is no direct relation between the unpaired LR–HR images in the training procedure, as illustrated in Algorithm 1. LR or HR input is trained in its own cycle, forward or backward. Loss functions in Section III-B are designed to make our Cycle-CNN to better recover the input image by alternately using the generators  $G_1$  and  $G_2$ . Therefore, although HR

data are used for unpaired training, such HR information may not be regarded as supervision information for SR like other supervised SR methods [20]–[22].

The structure of  $G_1$  refers to SRResNet [22]. Compared to SRResNet, we remove the batch normalization layer [40], which is proved to be not necessary in SR problems [41]. In addition, we adapt the 16 resblocks as the core structure of the network  $G_1$ . One resblock consists of two convolution layers, a rectified linear unit (ReLU) layer [42], and a pixel-wise addition layer. For the up-sampling layer, the proposed method uses subpixel layers [43] for up-sampling in  $G_1$ .

The structure of the down-sampling CNN  $G_2$  is opposite to  $G_1$ , and it contains five resblocks. In addition, for  $G_2$ , in order to downscale the image, we use two  $2 \times 2$ -average-pooling layers before resblock1. The following provides more implementation details of our network.

1) *Subpixel Layer*: Subpixel layer is an end-to-end up-sampling layer, which can up-sample the feature map by generating a plurality of channels by convolution and reshaping them. Assume that the size of the input feature map is  $H \times W \times C$  and the up-sample scale is  $s$ . The feature map is first convolved by a convolution kernel with a size of  $C \times k \times k \times s^2C$  ( $k = 3$  in our method), the output size

is  $H \times W \times s^2C$ , and then the output is reshaped to size  $sH \times sW \times C$  by the method named *shuffle* [43].

2) *Skip Connection*: Skip connection is a common strategy in SR tasks [21], [22], [41]. It can speed up convergence and improve the reconstruction results. Our proposed method has two kinds of skip connections. One is in resblock, connecting the input and output of resblock. The other connects the input of resblock1 and the output of the text convolution layer of resblock16 (resblock5 in  $G_2$ ).

### B. Losses of Cycle-CNN

Since we expect the cycle network to bring the generated image back to the original one, we use the cycle consistency loss. The consistency loss  $L_{cyc}$  is made up of forward consistency loss  $L_{cyc}^f$  and backward consistency loss  $L_{cyc}^b$  as

$$L_{cyc} = \lambda_1 L_{cyc}^f + \lambda_2 L_{cyc}^b \quad (2)$$

$$L_{cyc}^f = \frac{1}{N} \sum_{i=1}^N (\|G_2(G_1(\text{LR}\mathbf{x}_i)) - \text{LR}\mathbf{x}_i\|_2) \quad (3)$$

$$L_{cyc}^b = \frac{1}{N} \sum_{i=1}^N (\|G_1(G_2(\text{HR}\mathbf{y}_i)) - \text{HR}\mathbf{y}_i\|_2) \quad (4)$$

where  $\lambda_1$  and  $\lambda_2$  are weights of consistency loss,  $(\text{LR}\mathbf{x}_i, \text{HR}\mathbf{y}_i)$  are the unpaired LR/HR images, and  $i$  is the image index.

In the SR problem, Cycle-CNN needs the identity loss to make sure the input and output image content of the generated network ( $G_1$  and  $G_2$ ) is consistent. Furthermore, the identity loss can make the network more stable and easy to converge. There are two common losses used in the SR field: the pixel-wise loss and the content loss [44]. For the pixel-wise loss, we use L2 loss (i.e., MSE), and for the content loss, we use VGG loss based on the ReLU activation layers of the pretrained 19 layer VGG network [45]. The identity can act on  $G_1$  or  $G_2$ ; therefore, the identity loss has four kinds of forms, MSE-G1, MSE-G2, VGG-G1, and VGG-G2.

The MSE-G1 is calculated as

$$L_{\text{idt}}^{\text{MSE-G1}} = \frac{1}{N} \sum_{i=1}^N \|G_1(\text{HR}\mathbf{y}_i \downarrow_s) - \text{HR}\mathbf{y}_i\|_2 \quad (5)$$

where  $s$  represents bicubic down-sampling, and  $\text{HR}\mathbf{y}_i \downarrow_s$  is the down-sampled result of  $\text{HR}\mathbf{y}_i$ . In addition, if we use paired training for our Cycle-CNN (for comprehensive performance comparison with supervised models in Section IV-D3), i.e.,  $\text{HR}\mathbf{x}_i$  represents the ground truth HR image of  $\text{LR}\mathbf{x}_i$ , the identity loss will change to

$$L_{\text{idt}}^{\text{MSE-G1-P}} = \frac{1}{N} \sum_{i=1}^N \|G_1(\text{LR}\mathbf{x}_i) - \text{HR}\mathbf{x}_i\|_2. \quad (6)$$

The MSE-G2 is calculated as

$$L_{\text{idt}}^{\text{MSE-G2}} = \frac{1}{N} \sum_{i=1}^N \|G_2(\text{HR}\mathbf{y}_i) - \text{HR}\mathbf{y}_i \downarrow_s\|_2. \quad (7)$$

The VGG-G1 is calculated as

$$L_{\text{idt}}^{\text{VGG-G1}/m,n} = \frac{1}{N} \sum_{i=1}^N \|\phi_{m,n}(G_1(\text{LR}\mathbf{x}_i)) - \phi_{m,n}(\text{LR}\mathbf{x}_i)\|_2 \quad (8)$$

where  $\phi_{m,n}$  indicates the feature map of the  $n$ th convolution (after activation) before the  $m$ th maxpooling layer within the VGG19 network [22].

The VGG-G2 is calculated as

$$L_{\text{idt}}^{\text{VGG-G2}/m,n} = \frac{1}{N} \sum_{i=1}^N \|\phi_{m,n}(G_2(\text{HR}\mathbf{y}_i)) - \phi_{m,n}(\text{HR}\mathbf{y}_i)\|_2. \quad (9)$$

The experimental results of different losses are presented in Section IV-D.

The total loss of our Cycle-CNN is

$$L_{\text{total}} = \omega_1 L_{cyc} + \omega_2 L_{\text{idt}} \quad (10)$$

where  $\omega_1$  and  $\omega_2$  are the weights for linear combination.

### C. Training Details

Our goal is to train two generate functions  $G_1$  and  $G_2$ .  $G_1$  estimates the HR images  $\text{HR}I$  from a given LR input  $\text{LR}I$ . Relatively, for a given HR input image  $\text{HR}I$ ,  $G_2$  estimates its corresponding LR image  $\text{LR}I$  counterpart. Assume that the size of  $\text{LR}I$  is  $H \times W \times C$  and the scale factor is  $s$ , then the size of  $\text{HR}I$  is  $sH \times sW \times C$ . In our experiments, we only reconstruct the luminance channel of remote sensing images, therefore the value of  $C$  is 1. The parameters of  $G_1$  and  $G_2$  are  $\theta_{G1}$  and  $\theta_{G2}$ , respectively. In addition, to test the proposed model, only  $G_1$  is used.

In our proposed method, we adapt unpaired LR/HR remote sensing images to train the network.  $\text{LR}\mathbf{x}_i, i = 1, \dots, N$  are unpaired LR training images and  $\text{HR}\mathbf{y}_i, i = 1, \dots, N$  are unpaired HR training images.  $N$  is number of images in a minibatch. The proposed training approach is summarized in Algorithm 1.

---

#### Algorithm 1 Pseudocode of Training Cycle-CNN

---

**Require:** Unpaired LR/HR remote sensing training images

$(\text{LR}\mathbf{x}_i, \text{HR}\mathbf{y}_i), i = 1, \dots, N$

**Goal** The well-trained  $G_1, G_2$

1: Initialize  $\theta_{G1}, \theta_{G2}$

2: **repeat**

3:  $\text{HR}\mathbf{x}_i \leftarrow G_1(\text{LR}\mathbf{x}_i)$

4:  $\text{LR}\mathbf{x}'_i \leftarrow G_2(\text{HR}\mathbf{x}_i)$

5:  $L_{cyc}^f \leftarrow \text{MSE}(\text{LR}\mathbf{x}_i, \text{LR}\mathbf{x}'_i)$

6:  $\text{LR}\mathbf{y}'_i \leftarrow G_2(\text{HR}\mathbf{y}_i)$

7:  $\text{HR}\mathbf{y}'_i \leftarrow G_1(\text{LR}\mathbf{y}'_i)$

8:  $L_{cyc}^b \leftarrow \text{MSE}(\text{HR}\mathbf{y}_i, \text{HR}\mathbf{y}'_i)$

9:  $L_{cyc} \leftarrow \lambda_1 L_{cyc}^f + \lambda_2 L_{cyc}^b$

10:  $\text{HR}\mathbf{y}_i^{\text{down}} \leftarrow G_1(\text{HR}\mathbf{y}_i \downarrow_s)$

11:  $L_{\text{idt}} \leftarrow \text{MSE}(\text{HR}\mathbf{y}_i^{\text{down}}, \text{HR}\mathbf{y}_i)$

12:  $L_{\text{total}} \leftarrow \omega_1 L_{cyc} + \omega_2 L_{\text{idt}}$

13: ADAM-optimizer( $L_{\text{total}}$ , variable= ( $\theta_{G1}, \theta_{G2}$ ))

14: **until** Reach maximum iteration of minibatch updating

15:

16: **function** CYCLE-CNN-TEST( $X_{LR}$ )

17:  $X_{HR} \leftarrow G_1(X_{LR})$

18: **return**  $X_{HR}$

19: **end function**

---

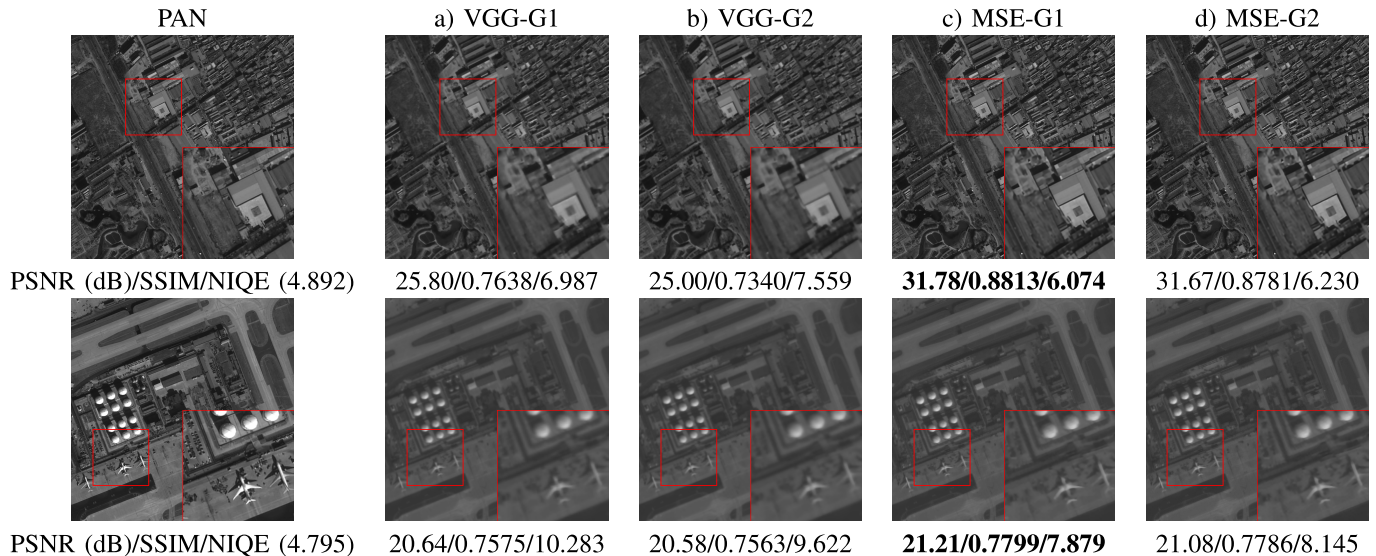


Fig. 2. SR results of different loss functions. The first rows are results of bicubic down-sampled PAN images, and the second row shows the results of Y channels converted by MS images.

Due to the memory limitation of GPUs, we crop  $96 \times 96$  regions from  $256 \times 256$  images as LR patches, and crop  $384 \times 384$  patches from  $1024 \times 1024$  PAN images as HR patches. The minibatch is 8. We train our model with Adam optimizer [46] by setting the parameters  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . The learning rate is initialized as  $10^{-4}$  and decreases by factor ten at half of the training process. The iteration of minibatch updating is  $1.8 \times 10^5$ . We set the consistency loss weights  $\lambda_1 = 1$ ,  $\lambda_2 = 1$  and set the total loss weights  $\omega_1 = 2$ ,  $\omega_2 = 1$ . We update the parameters of  $G_1$  and  $G_2$  at the same time.

#### IV. EXPERIMENTS AND RESULTS

##### A. Experiment Configuration

Our experiments are carried out under the following software and hardware conditions. The CNN-based methods in our experiments, i.e., SRCNN [20], VDSR [21], SRResNet [22], ZSSR [33], and our proposed method are performed in the GPU environment. We implement other methods including bicubic [12] and SelfEx [32] in the CPU environment.

1) *GPU Environment*: The GPU environment is with Inter (R) Xeon (R) CPU E5-2630 V4 at 2.20 GHz and 16 GB DDR4 RAM. We train and test the network on a NVIDIA GeForce GTX 1080Ti with a memory of 11 GB. The operating system is Linux Ubuntu 16.04.10  $\times 64$ . We implement our models with TensorFlow1.9.0 framework. In addition, we use CUDA9 for GPU calculation and TensorLayer API [47] for the construction of our network models.

2) *CPU Environment*: The CPU environment is composed of Intel (R) Core (TM) i7-8750H at 2.20 GHz CPU and DDR4 2666 RAM with a capacity of 8 GB. The operating system is Windows10  $\times 64$ , the experiments are run on MATLAB R2016a.

Our codes and data will be publicly available through our website: <https://github.com/haopzhang/CycleCNN>.

##### B. Data Set

1) *GaoFen-2* [48]: We collect 720 image pairs from the GaoFen-2 satellite for experiments, including panchromatic (PAN) band images and the corresponding multispectral (MS) band images. PAN images are regarded as HR images with spatial resolution 1 m/pixel and size of  $1024 \times 1024$ . The size of MS images is  $256 \times 256$  and their spatial resolution is 4 m/pixel. We convert the first three bands of MS (i.e., the blue, green, and red channels) to YCbCr color space, and regard Y channels as the LR images for unpaired training [2]. Thus, the SR scale factor is 4. For supervised SR networks, we down-sample PAN images to get LR images for pairwise training. We randomly select three image pairs for validation and five pairs for testing. The remaining 712 pairs are used for training. Specifically, the testing set contains eight images of spatial resolution 4 m/pixel, named ALL8. For a fair comparison, five testing images are down-sampled from PAN images (PAN5), and three of them are Y channels converted by MS images (MS3). In addition, to analyze the robustness against noise and blur, we also add Gaussian blur and Gaussian white noise of standard deviation ten to the testing images. Their corresponding five PAN images are used to calculate the popular full-reference indexes PSNR and SSIM for SR performance evaluation. It should be noticed that the PAN and MS images were respectively captured by two different cameras both aboard the GaoFen-2 satellite.

2) *UC Merced* [49]: UC Merced land use data set is an extensive manually labeled ground truth data set. The data set consists of images of 21 classes with a spatial resolution of one foot. Each class contains 100 images with a size of  $256 \times 256$  pixels. In the experiments, we use it to perform a quantitative evaluation with comparison to unsupervised SR methods. We select one image per class for testing. To make experimental results comparable, we use the same testing images as [35] in the following 12 classes: agricultural, agricultural2, airplane, baseball, bridge, circular-farmland, harbor,

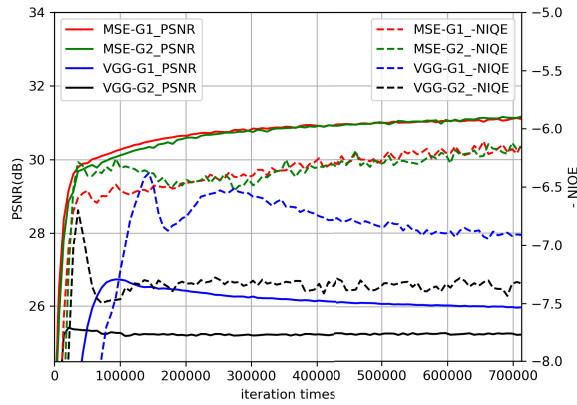


Fig. 3. Training performance of different identity loss for down-sampled PAN GaoFen-2 valid images. The index is average PSNR (dB)/-NIQE of reconstructed bicubic down-sampled PAN GaoFen-2 valid images.

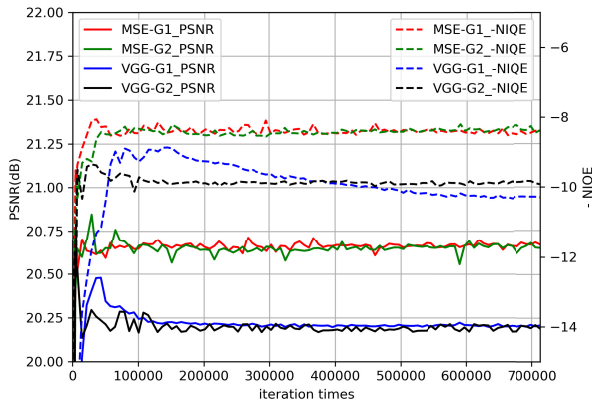


Fig. 4. Training performance of different identity loss for Y channels converted by MS GaoFen-2 valid images. The index is average PSNR (dB)/-NIQE of reconstructed Y channels converted by MS GaoFen-2 valid images.

industry, intersection, parking, residential, and road. It means that 21 images are in our testing set while the results of 12 of them are reported in this article. We also randomly select three images per class for validation. The remaining 96 images per class are used for training.

### C. Evaluation Index

Peak signal-to-noise ratio (PSNR) is widely used to measure the reconstruction quality. PSNR is defined via the maximum possible pixel value (denote as  $L$ ) and the mean squared error (MSE) between images.  $L = 255$  in most of our experiments. Only in Table IV,  $L$  is the maximum pixel value of single test image. Given the ground truth  $X$  and constructed image  $X_{SR}$ , and  $N$  is the total pixels of both of the images, the MSE and the PSNR are defined as follows:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N \|X(i) - X_{SR}(i)\|_2 \quad (11)$$

$$\text{PSNR} = 10 \log_{10} \frac{L^2}{\text{MSE}}. \quad (12)$$

The structural similarity index (SSIM) [50] is used for measuring the structural similarity between images, based on

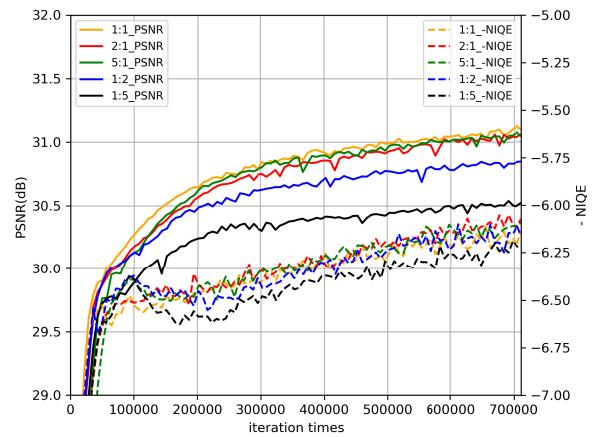


Fig. 5. Training performance of different weights proportion of consistency loss ( $L_{cyc}^f : L_{cyc}^b$ ). The index is average PSNR (dB)/-NIQE of reconstructed bicubic down-sampled PAN GaoFen-2 valid images.

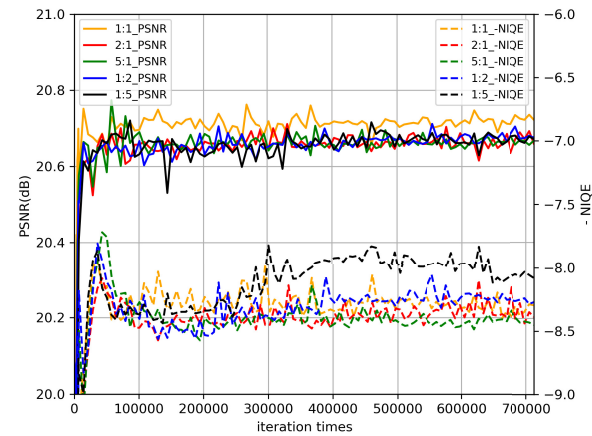


Fig. 6. Training performance of different weights proportion of consistency loss ( $L_{cyc}^f : L_{cyc}^b$ ). The index is average PSNR (dB)/-NIQE of reconstructed Y channel converted by MS GaoFen-2 valid images.

three relatively independent comparisons, luminance, contrast, and structure. For the ground truth  $X$  and constructed image  $X_{SR}$ ,  $\mu_X$ ,  $\sigma_X$  represent the mean and the standard deviation of  $X$ ,  $\mu_{X_{SR}}$ ,  $\sigma_{X_{SR}}$  represent the mean and the standard deviation of  $X_{SR}$ , and  $\sigma_{XX_{SR}}$  is the covariance between  $X$  and  $X_{SR}$ . The SSIM is defined as

$$\text{SSIM}(X, X_{SR}) = \frac{(2\mu_X \mu_{X_{SR}} + C_1)(\sigma_{XX_{SR}} + C_2)}{(\mu_X^2 + \mu_{X_{SR}}^2 + C_1)(\sigma_X^2 + \sigma_{X_{SR}}^2 + C_1)} \quad (13)$$

where  $C_1 = (k_1 L)^2$  and  $C_2 = (k_2 L)^2$  are constants for avoiding instability.

Spectral angle mapper (SAM) [51] calculates the angle between the spectra, and determines the similarity between the two spectra. The *smaller* SAM means the two images are more similar.

$$\text{SAM}(X, X_{SR}) = \frac{1}{N} \sum_{i=1}^N \arccos \frac{X(i)X_{SR}(i)}{\|X(i)\| \|X_{SR}(i)\|}. \quad (14)$$

Erreur relative globale adimensionnelle de synthese (ERGAS) [52] evaluates the quality of all the bands of remote

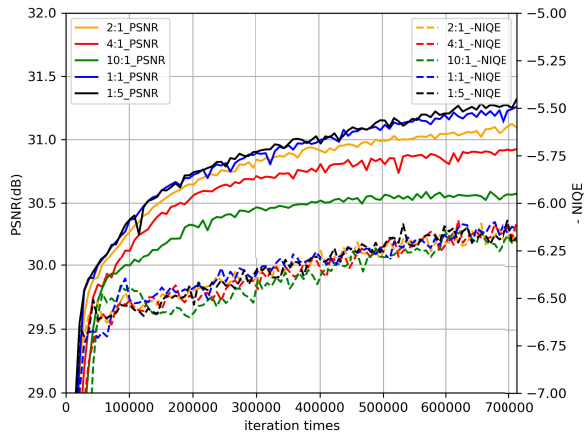


Fig. 7. Training performance of different weights proportion of consistency loss ( $L_{cyc} : L_{idt}$ ). The index is average PSNR (dB)/-NIQE of reconstructed bicubic down-sampled PAN GaoFen-2 valid images.

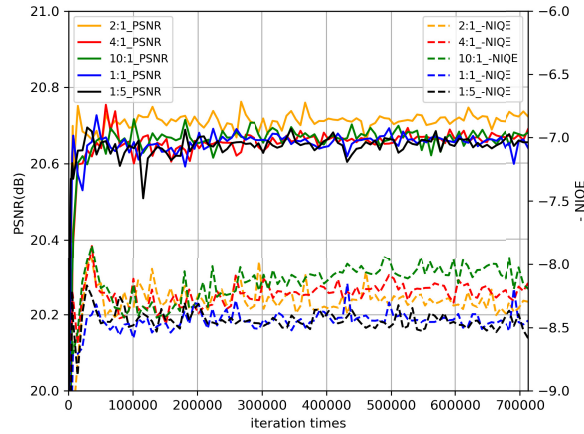


Fig. 8. Training performance of different weights proportion of consistency loss ( $L_{cyc} : L_{idt}$ ). The index is average PSNR (dB)/-NIQE of reconstructed Y channel converted by MS GaoFen-2 valid images.

sensing image. Its *smaller* value stands for *better* image quality

$$\text{ERGAS}(X, X_{SR}) = 100 \frac{l}{h} \sqrt{\frac{1}{N_{\text{bands}}} \sum_{i=1}^{N_{\text{bands}}} \left( \frac{\text{RMSE}(X(i), X_{SR}(i))}{\bar{X}(i)} \right)^2} \quad (15)$$

where  $N_{\text{bands}}$  represents the numbers of bands, and  $h$  and  $l$  represent the resolution of HR and LR, respectively.

No-reference-based metrics have been proposed to predict the image quality without ground truth. NIQE [53] is a popular metric used for evaluating the image quality of SR. NIQE uses three types of low-level statistical features in both spatial and frequency domains to quantify SR artifacts, and learn a two-stage regression model to calculate the scores of SR images. The *lower* NIQE score means the *better* image quality.

#### D. Results and Discussion

1) *Type of Identity Loss Functions*: Table I and Fig. 2 show the experimental results of different identity loss functions of our Cycle-CNN method, and Figs. 3 and 4 present the

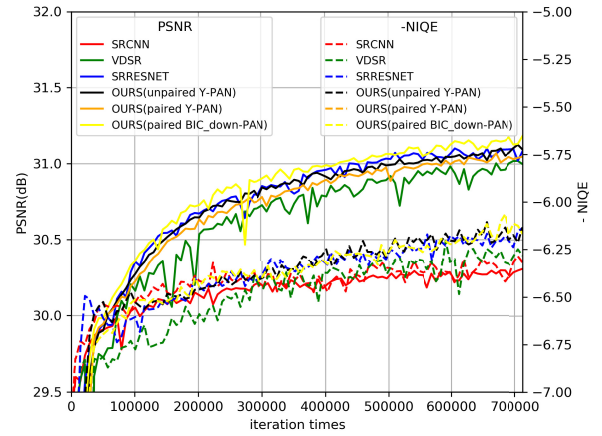


Fig. 9. Training performance of different CNN-based networks for down-sampled PAN GaoFen-2 valid images. The index is average PSNR (dB)/-NIQE of reconstructed bicubic down-sampled PAN GaoFen-2 valid images.

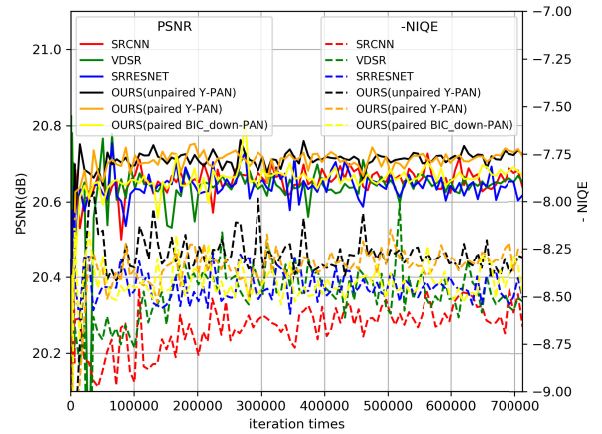


Fig. 10. Training performance of different CNN-based networks for Y channels converted by MS GaoFen-2 valid images. The index is average PSNR (dB)/-NIQE of reconstructed Y channels converted by MS GaoFen-2 valid images.

training performance of four identity losses in reconstruct bicubic down-sampled PAN GaoFen-2 valid images and Y channel converted by the MS.

According to the experimental results, MSE-G1 and MSE-G2 have much better performance than VGG-G1 and VGG-G2. It indicates that VGG identity loss is not suitable for our Cycle-CNN. VGG identity loss aims to make the details of two images closer; however, the purpose of SR is to restore the details of LR images. Comparing MSE-G1 and MSE-G2, MSE-G1 has shown better results in the test data set. It can be observed that the average PSNR of MSE-G1 is higher than MSE-G2 by 0.14 dB and the average NIQE of MSE-G1 is lower than MSE-G2 by 0.223. In conclusion, we choose MSE-G1 as the identity loss in our method.

2) *Weights of Loss Functions*: In order to explore the impact on different loss function weights in our method on the super-resolved results, we design two groups of tests. One is the different proportion of cycle consistency loss, i.e.,  $L_{cyc}^f : L_{cyc}^b$ , whose weights are  $\lambda_1$  and  $\lambda_2$ . The other is different  $\omega_1 : \omega_2$  values, representing  $L_{cyc} : L_{idt}$ . Table II and Figs. 5–8 show our experimental results. First, in terms

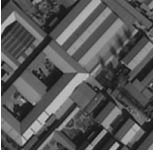


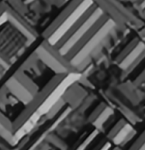


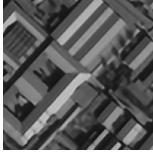







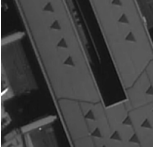

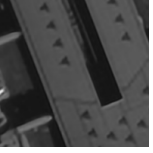
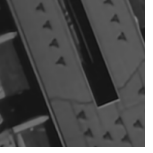
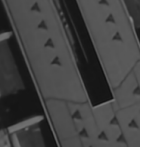
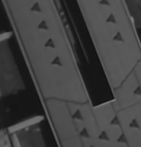









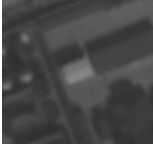
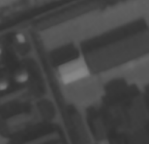
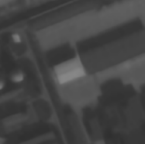
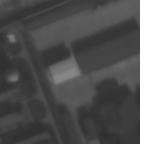
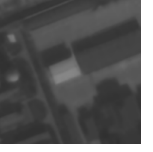
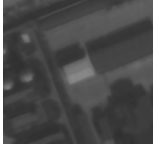
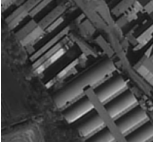






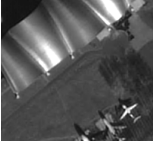
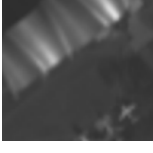
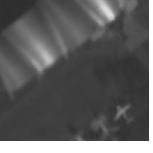
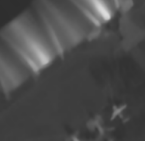
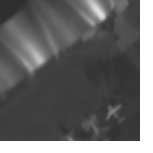
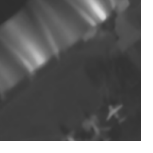
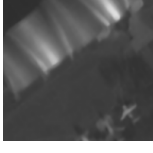
| a) PAN  | b) Bi-cubic   | c) SRCNN  | d) VDSR   | e) SRResNet  | f) Ours(paired)   | g) Ours(unpaired)   |
|---|---|---|---|--|---|---|
|    |    |    |    |    |    |    |
| PSNR (dB)/SSIM<br>NIQE=5.070  | 27.57/0.7807<br>7.324   | 29.87/0.8425<br>6.810   | 30.80/0.8654<br>6.273   | 31.06/0.8697<br>6.203  | <b>31.47/0.8784</b><br><b>6.047</b>   | 31.25/0.8749<br>6.060   |
|    |    |    |    |    |    |    |
| PSNR (dB)/SSIM<br>NIQE=4.795  | 30.02/0.8700<br>7.177   | 31.98/0.9019<br>6.405   | 32.77/0.9148<br>6.067   | 32.98/0.9173<br>5.857  | <b>33.25/0.9217</b><br><b>5.636</b>   | 33.07/0.9193<br>5.701   |
|    |    |    |    |    |    |    |
| PSNR (dB)/SSIM<br>NIQE=5.843  | 30.79/0.9101<br>7.418   | 33.54/0.9357<br>6.384   | 34.74/0.9476<br>6.338   | 35.36/0.9504<br>5.959  | <b>35.97/0.9535</b><br>5.903  | 35.58/0.9518<br><b>5.631</b>  |
|   |   |   |   |   |   |   |
| PSNR (dB)/SSIM<br>NIQE=5.396  | 31.36/0.8577<br>7.662   | 32.84/0.8885<br>6.720   | 33.45/0.8984<br>6.772   | 33.71/0.9017<br>6.685  | <b>33.96/0.9060</b><br><b>6.501</b>   | 33.73/0.9022<br>6.511   |
|  |  |  |  |  |  |  |
| PSNR (dB)/SSIM<br>NIQE=4.795  | 22.14/0.8536<br>9.167   | 22.16/0.8570<br>8.555   | 22.16/0.8567<br>8.296   | 22.16/0.8569<br>8.179  | 22.20/0.8573<br>8.212   | <b>22.33/0.8579</b><br><b>7.879</b>   |
|  |  |  |  |  |  |  |
| PSNR (dB)/SSIM<br>NIQE=5.070  | 20.41/0.6234<br>9.199   | 20.44/0.6249<br>8.436   | 20.44/0.6257<br>8.394   | 20.44/0.6249<br>8.264  | 20.45/0.6257<br>8.450   | <b>20.50/0.6262</b><br><b>8.084</b>   |
|  |  |  |  |  |  |  |
| PSNR (dB)/SSIM<br>NIQE=5.396  | 22.35/0.7157<br>9.271   | 22.35/0.7157<br>8.261   | 22.35/0.7159<br>8.375   | 22.36/0.7162<br>8.290  | 22.35/ <b>0.7165</b><br>8.322   | <b>22.38/0.7160</b><br><b>8.008</b>   |

Fig. 11. SR results of GaoFen-2. The first four rows are results of bicubic down-sampled PAN images, and the last three rows show the results of Y channels converted by MS images. All pictures shown here are  $200 \times 200$  areas cropped from  $1024 \times 1024$  reconstructed HR images.

of different  $\lambda_1 : \lambda_2$ , Fig. 5 shows that with the increase of  $\lambda_2$ , the performance becomes worse in reconstruct bicubic down-sampled images. While Fig. 6 indicates that decreasing

specific value of  $\lambda_1 : \lambda_2$  can improve the results of Y channels converted by MS images. Second, regarding  $\omega_1 : \omega_2$ , Fig. 7 demonstrates that lower  $\omega_1 : \omega_2$  value can bring better results



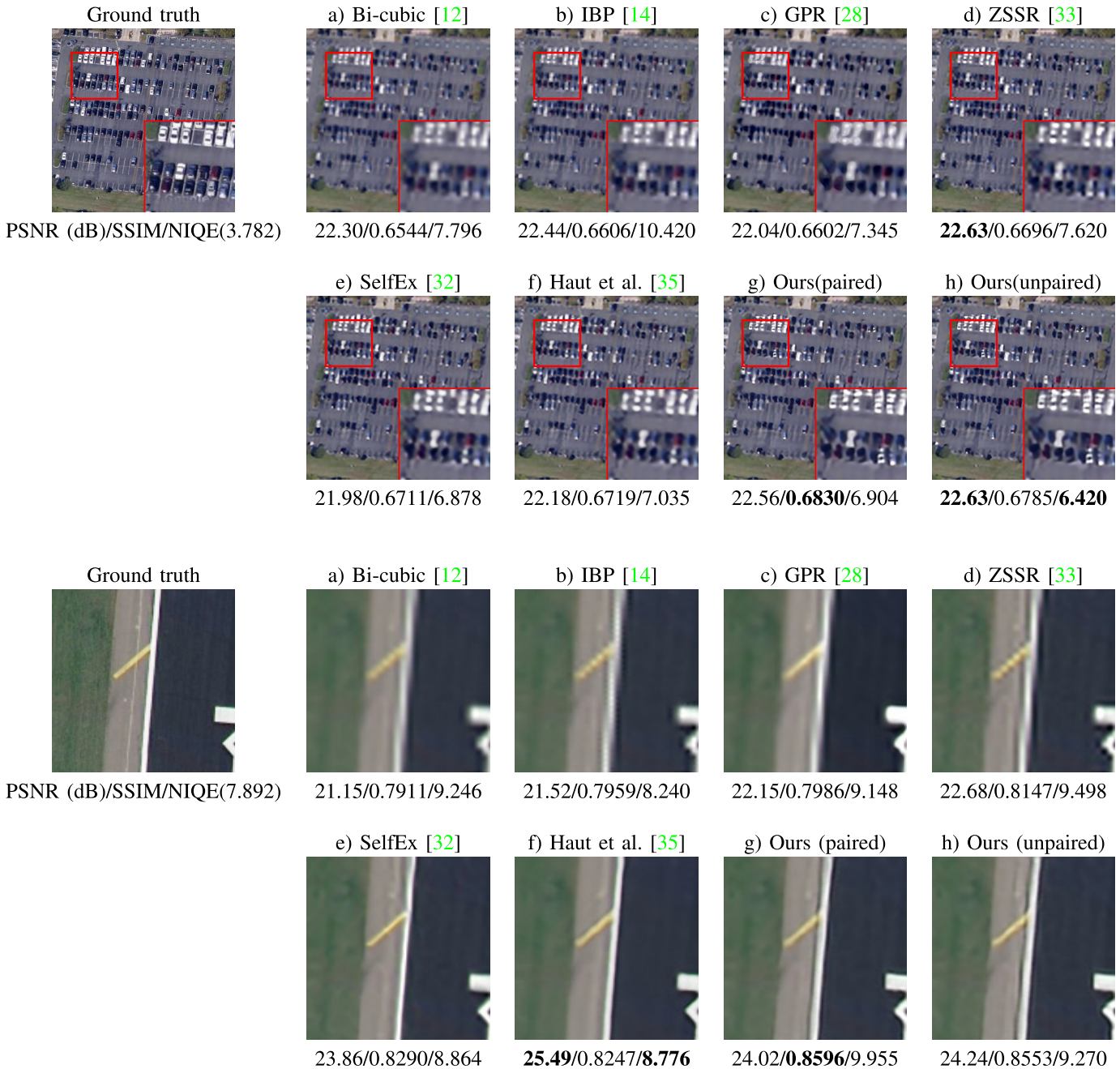


Fig. 12. SR results of parking and road with a  $4\times$  scale factor on the UC Merced data set. In our method, we only reconstruct the luminance channel in YCbCr color space by the network, while the Cb and Cr channels are reconstructed via bicubic interpolation.

TABLE I

COMPARISON BETWEEN DIFFERENT LOSS FUNCTIONS OF OUR METHODS. (PSNR (dB)/SSIM/NIQE). PAN5 REPRESENTS FIVE TESTING IMAGES DOWN-SAMPLED FROM PAN IMAGES, AND MS3 ARE Y CHANNELS CONVERTED BY THREE MS IMAGES. ALL8 MEANS ALL OF THE EIGHT IMAGES OF SPATIAL RESOLUTION 4 M/PIXEL, I.E., PAN5+MS3

| test images type | VGG-G1             | VGG-G2             | MSE-G1                    | MSE-G2             |
|------------------|--------------------|--------------------|---------------------------|--------------------|
| ALL8             | 24.62/0.7757/8.079 | 24.17/0.7610/8.205 | <b>28.54/0.8453/6.587</b> | 28.40/0.8427/6.820 |
| PAN5             | 26.93/0.8027/7.074 | 26.25/0.7800/7.565 | <b>32.85/0.8996/6.042</b> | 32.72/0.8962/6.149 |
| MS3              | 20.75/0.7306/9.754 | 20.69/0.7293/9.273 | <b>21.35/0.7547/7.495</b> | 21.22/0.7535/7.939 |

in super-resolved bicubic down-sampled images. Conversely, when reconstructing Y channels, with the increase of  $\omega_1 : \omega_2$  value, the reconstructed images have lower NIQE index, which means higher image quality.

3) *Comparison With State-of-the-Art Supervised SR Methods*: Taking bicubic interpolation as the baseline, we compare the proposed method with other state-of-the-art supervised CNN-based SR methods such as SRCNN [20], VDSR [21],

TABLE II

COMPARISON BETWEEN DIFFERENT LOSS FUNCTION WEIGHTS OF OUR METHODS. (PSNR (dB) /SSIM / NIQE).  $\omega_1 : \omega_2$  IS THE PROPORTION OF  $L_{CYC}$  AND  $L_{IDT}$ , AND  $\lambda_1 : \lambda_2$  IS THE PROPORTION OF  $L_{CYC}^f$  AND  $L_{IDT}^b$ . THE ABOVE PART SHOWS THE EXPERIMENT OF CONSISTENCY LOSS WEIGHTS AND THE BELOW PART SHOWS THE EXPERIMENT OF TOTAL LOSS WEIGHTS. **RED** COLOR INDICATES THE BEST PERFORMANCE AND **BLUE** COLOR REFERS THE SECOND BEST OF ALL THE WEIGHTS PROPORTION

| $\omega_1 : \omega_2, \lambda_1 : \lambda_2$ | ALL8         |               |              | PAN5         |               |              | MS3          |               |              |
|--|--------------|---------------|--------------|--------------|---------------|--------------|--------------|---------------|--------------|
|  | PSNR         | SSIM          | NIQE         | PSNR         | SSIM          | NIQE         | PSNR         | SSIM          | NIQE         |
| 2:1,1:1                                      | <b>28.54</b> | <b>0.8453</b> | <b>6.587</b> | 32.85        | <b>0.8996</b> | <b>6.042</b> | <b>21.35</b> | <b>0.7547</b> | 7.495        |
| 2:1,2:1                                      | 28.35        | 0.8423        | 6.836        | 32.62        | 0.8951        | 6.184        | 21.22        | 0.7542        | 7.923        |
| 2:1,5:1                                      | 28.35        | 0.8423        | 6.859        | 32.62        | 0.8951        | 6.189        | 21.24        | 0.7542        | 7.978        |
| 2:1,1:2                                      | 28.23        | 0.8408        | 6.705        | 32.41        | 0.8925        | 6.263        | 21.25        | <b>0.7548</b> | 7.443        |
| 2:1,1:5                                      | 27.98        | 0.8372        | 6.705        | 32.02        | 0.8868        | 6.326        | 21.24        | 0.7545        | <b>7.335</b> |
| 4:1,1:1                                      | 28.23        | 0.8405        | <b>6.679</b> | 32.40        | 0.8923        | 6.196        | <b>21.28</b> | 0.7542        | 7.486        |
| 10:1,1:1                                     | 28.00        | 0.8375        | 6.764        | 32.04        | 0.8871        | 6.369        | 21.26        | <b>0.7548</b> | <b>7.421</b> |
| 1:1,1:1                                      | 28.49        | 0.8441        | 6.826        | <b>32.87</b> | 0.8986        | <b>6.176</b> | 21.19        | 0.7536        | 7.910        |
| 1:5,1:1                                      | <b>28.56</b> | <b>0.8451</b> | 6.866        | <b>32.97</b> | <b>0.8998</b> | 6.197        | 21.22        | 0.7539        | 7.982        |

TABLE III

COMPARISON BETWEEN OUR METHOD AND OTHER STATE-OF-THE-ART CNN-BASED METHODS. BICUBIC IS THE BASELINE. \* REPRESENTS THAT THE NETWORK USES UNPAIRED LR-HR FOR TRAINING. B DENOTES THAT THE LR TRAINING SET IS BICUBIC DOWN-SAMPLED FROM PAN IMAGES; COMPAREATIVELY, Y DENOTES THAT IT USES Y CHANNELS CONVERTED BY MS AS LR TRAINING SET. ALL THE METHODS ADOPT PAN IMAGES AS HR TRAINING SET. PAN5 REPRESENTS FIVE TESTING IMAGES DOWN-SAMPLED FROM PAN IMAGES, AND MS3 ARE Y CHANNELS CONVERTED BY THREE MS IMAGES. ALL8 MEANS ALL OF THE EIGHT IMAGES OF SPATIAL RESOLUTION 4 M/PIXEL, I.E., PAN5+MS3. **RED** COLOR INDICATES THE BEST PERFORMANCE AND **BLUE** COLOR REFERS THE SECOND BEST

| Test image | Method          | None          |               |              | Blur          |               |              | Noise         |               |              | Blur+Noise    |               |              |
|------------|-----------------|---------------|---------------|--------------|---------------|---------------|--------------|---------------|---------------|--------------|---------------|---------------|--------------|
|            |                 | PSNR          | SSIM          | NIQE         | PSNR          | SSIM          | NIQE         | PSNR          | SSIM          | NIQE         | PSNR          | SSIM          | NIQE         |
| ALL8       | Bi-cubic [12]   | 26.41         | 0.8045        | 8.214        | 25.56         | 0.7806        | 10.414       | 26.00         | 0.7773        | 7.703        | 25.14         | <b>0.7597</b> | 8.351        |
|            | SRCNN-B [20]    | 27.69         | 0.8312        | 7.106        | 26.49         | 0.8030        | <b>7.808</b> | 26.64         | 0.7813        | 8.077        | 25.66         | 0.7454        | 8.393        |
|            | VDSR-B [21]     | 28.30         | 0.8415        | 6.976        | 26.64         | 0.8072        | 8.096        | 27.03         | <b>0.7884</b> | 7.823        | <b>25.82</b>  | 0.7538        | 8.101        |
|            | SRResNet-B [22] | 28.48         | 0.8436        | 6.754        | <b>26.78</b>  | 0.8096        | 8.437        | 26.94         | 0.7839        | 7.563        | 25.80         | 0.7517        | <b>7.807</b> |
|            | SRResNet-Y      | 16.64         | 0.6383        | 12.649       | 16.64         | 0.6381        | 16.566       | 16.65         | 0.6010        | 12.422       | 16.66         | 0.6008        | 12.482       |
|            | SRResNet*-Y     | 16.79         | 0.6386        | 16.813       | 16.81         | 0.6396        | 16.783       | 16.80         | 0.6023        | 11.752       | 16.81         | 0.6023        | 17.493       |
|            | Ours-B          | <b>28.67</b>  | <b>0.8470</b> | <b>6.632</b> | 26.76         | 0.8097        | 8.541        | <b>27.14</b>  | 0.7866        | 7.227        | <b>25.82</b>  | 0.7500        | 7.965        |
|            | Ours-Y          | 28.52         | 0.8420        | 7.116        | <b>26.78</b>  | <b>0.8010</b> | 8.321        | <b>27.13</b>  | 0.7849        | <b>7.057</b> | 25.80         | 0.7501        | 7.976        |
| Ours*-Y    | <b>28.54</b>    | <b>0.8453</b> | <b>6.587</b>  | <b>26.84</b> | <b>0.8109</b> | <b>7.636</b>  | 27.12        | <b>0.7896</b> | <b>6.684</b>  | <b>25.93</b> | <b>0.7550</b> | <b>7.632</b>  |              |
| PAN5       | Bi-cubic        | 29.53         | 0.8368        | 7.352        | 28.60         | 0.8157        | 8.043        | 28.74         | 0.8321        | 7.419        | 27.70         | 0.8106        | 8.014        |
|            | SRCNN-B         | 31.57         | 0.8778        | 6.666        | 30.09         | 0.8520        | 6.892        | 28.75         | 0.8421        | 6.877        | 28.75         | 0.8421        | 7.150        |
|            | VDSR-B          | 32.55         | 0.8943        | 6.382        | 30.33         | 0.8584        | <b>6.664</b> | 28.98         | 0.8483        | 6.477        | 28.98         | 0.8483        | <b>6.796</b> |
|            | SRResNet-B      | 32.83         | 0.8977        | 6.199        | <b>30.54</b>  | 0.8623        | 7.089        | 30.57         | <b>0.8865</b> | <b>6.157</b> | <b>29.06</b>  | <b>0.8509</b> | <b>6.765</b> |
|            | SRResNet-Y      | 16.97         | 0.6278        | 12.914       | 17.00         | 0.6278        | 12.215       | 16.98         | 0.6280        | 12.823       | 17.02         | 0.6279        | 12.777       |
|            | SRResNet*-Y     | 17.08         | 0.6258        | 15.461       | 17.10         | 0.6270        | 14.759       | 17.10         | 0.6266        | 15.528       | 17.12         | 0.6276        | 15.830       |
|            | Ours-B          | <b>33.12</b>  | <b>0.9027</b> | <b>5.974</b> | 30.50         | 0.8623        | 7.794        | <b>30.70</b>  | <b>0.8909</b> | <b>6.178</b> | 29.01         | <b>0.8510</b> | 7.139        |
|            | Ours-Y          | 32.83         | 0.8945        | 6.233        | 30.53         | <b>0.8630</b> | 7.073        | <b>30.63</b>  | 0.8825        | 6.382        | 29.00         | 0.8500        | 7.429        |
| Ours*-Y    | <b>32.85</b>    | <b>0.8996</b> | <b>6.042</b>  | <b>30.59</b> | <b>0.8641</b> | <b>6.220</b>  | 30.52        | <b>0.8909</b> | 6.294         | <b>29.07</b> | 0.8508        | 7.178         |              |
| MS3        | Bi-cubic        | 21.19         | 0.7506        | 9.651        | 20.51         | 0.7217        | 14.364       | 21.11         | <b>0.7027</b> | <b>8.177</b> | <b>20.88</b>  | <b>0.6709</b> | <b>8.912</b> |
|            | SRCNN-B         | 21.23         | 0.7534        | 7.840        | 20.50         | 0.7213        | <b>9.333</b> | <b>21.34</b>  | <b>0.6354</b> | 10.076       | 20.50         | 0.5842        | 10.464       |
|            | VDSR-B          | 21.23         | 0.7535        | 7.965        | 20.49         | 0.7219        | 10.481       | 21.25         | 0.6248        | 10.065       | 20.54         | <b>0.5962</b> | 10.277       |
|            | SRResNet-B      | 21.23         | 0.7535        | <b>7.678</b> | 20.51         | 0.7219        | 10.684       | 21.23         | 0.6130        | 9.907        | 20.48         | 0.5840        | 9.543        |
|            | SRResNet-Y      | 16.09         | 0.6559        | 12.206       | 16.04         | 0.6555        | 23.816       | 16.11         | 0.5561        | 11.752       | 16.06         | 0.5557        | 11.992       |
|            | SRResNet*-Y     | 16.30         | 0.6599        | 19.066       | 16.32         | 0.6607        | 20.539       | 16.29         | 0.5593        | 18.875       | 16.31         | 0.5602        | 20.266       |
|            | Ours-B          | 21.25         | 0.7541        | 7.731        | <b>20.53</b>  | <b>0.7220</b> | 11.132       | 21.23         | 0.6127        | 8.975        | 20.50         | 0.5815        | 9.343        |
|            | Ours-Y          | <b>21.33</b>  | <b>0.7545</b> | 8.587        | <b>20.53</b>  | <b>0.7220</b> | 10.402       | 21.29         | 0.6222        | 8.183        | 20.46         | 0.5835        | 8.887        |
| Ours*-Y    | <b>21.35</b>    | <b>0.7547</b> | <b>7.495</b>  | <b>20.60</b> | <b>0.7221</b> | <b>9.998</b>  | <b>21.44</b> | 0.6208        | <b>7.866</b>  | <b>20.68</b> | 0.5955        | <b>8.388</b>  |              |

and SRResNet [22]. We individually train our Cycle-CNN network and SRResNet (only  $G_1$  of our network) using both paired and unpaired data for comprehensive performance comparison.

Table III and Fig. 11 show the comparison results of our method with other state-of-the-art SR methods, and Figs. 9 and 10 shows the performance of our SR module with different training iterations. It can be seen that our Cycle-CNN can achieve better results than state-of-the-art supervised methods, and have better robustness against blur and noise. Particularly, our pairwise-trained Cycle-CNN may get better SR results

for down-sampled PAN data, but worse for real Y channel data. In contrast, our nonpairwise-trained Cycle-CNN has better results in reconstructing Y channels. This validates the role of nonpairwise training. It should be noticed that for real remote sensing data with high degradation (i.e., blur and noise in Table III), traditional bicubic interpolation may perform better for  $\times 4$  SR. It shows that image degradation may affect learning-based SR methods a lot when the SR scale is high. Furthermore, compared to only one generative network, the Cycle-CNN structure increases the average PSNR by 0.19, indicates that our Cycle-CNN is useful for constructing bicubic

TABLE IV

COMPARISON BETWEEN OUR METHOD AND OTHER UNSUPERVISED METHODS ON THE UC MERCED [49]. AVERAGE RESULTS OF 12 IMAGES ARE REPORTED. RESULTS WITH \* ARE REFERRED FROM [35]

| Method                  | Time (s)    | PSNR          | SSIM          | SAM           | ERGAS         |
|-------------------------|-------------|---------------|---------------|---------------|---------------|
| Bi-cubic [12]           | <b>0.01</b> | 23.59         | 0.7147        | 0.0233        | 4.913         |
| IBP [14]                | 1.370       | 23.22         | 0.6792        | 0.0242        | 4.961         |
| GPR [28]                | 109.38      | 22.88         | 0.7170        | 0.0324        | 5.196         |
| SelfEx [32]             | 22.485      | 24.73         | 0.6830        | 0.0237        | 4.386         |
| ZSSR [33]               | 10.167      | 24.26         | 0.7369        | 0.0237        | 4.546         |
| Haut <i>et al.</i> [35] | 156.71*     | <b>25.21*</b> | 0.6776*       | 0.0236*       | <b>4.193*</b> |
| Ours (unpaired)         | 1.105       | 24.33         | <b>0.7456</b> | <b>0.0231</b> | 4.459         |

down-sampled remote sensing images. Since supervised CNN-based networks are designed for using paired images to train, the super-resolved results provided by supervised methods perform poorly when the training data are unpaired or are not matched strictly, such as paired Y-PAN. Our method can be applied to paired or unpaired training data and achieves good reconstruction results for both bicubic down-sampled images and real remote sensing images.

4) *Comparison With State-of-the-Art Unsupervised SR Methods:* Table IV shows the comparison results with other unsupervised methods. It can be seen that except bicubic, our method spends the least time in reconstructing images. Methods based on image iteration such as [28] and [35] averagely consume more than 100 s to complete the reconstruction process. Other methods such as ZSSR [33] and SelfEx [32] also consume more than 10s because they need extracting features and training on input images. On the other hand, Haut *et al.* [35] obtained the best PSNR result as 25.21 and the best ERGAS result as 4.193 on the UC Merced data set, while our nonpairwise-trained method achieves the best SSIM value 0.7456 and SAM value 0.0231.

Fig. 12 shows the SR results of “parking” and “road” in the UC Merced data set. In Fig. 12, we can be observed that our method can obtain sharper edges and richer details, such as the parking line and the glasses on the car. In addition, the super-resolved images of our method have better clearness and visualization than other state-of-the-art unsupervised SR methods. In image “road,” Haut *et al.* [35] achieved the highest PSNR and the best-reconstructed quality for lines in the road, whereas our method can also restore the edges of lines. According to the results, we can conclude that considering the trade-off between super-resolved image quality and time costs, our proposed method has obvious advantages compared to unsupervised SR approaches shown in Table IV.

## V. CONCLUSION

In this article, we have proposed a nonpairwise-trained SR network named Cycle-CNN for remote sensing images. We set two modules in our network. The first module is used to map LR images to HR images, i.e., SR, while the second module maps HR images back to LR images, like down-sampling. According to the comparison of four types of identity loss and nine kinds of weights proportions, we can conclude that MSE-G1 is the best identity loss. Experimental results on

GaoFen-2 satellite images demonstrate that our proposed method achieves state-of-the-art SISR results and good robustness against blur and noise. Furthermore, our proposed method performs better while reconstructing multispectral band images of real remote sensing satellites. Tests on the UC Merced data set demonstrate that our proposed method can achieve a better trade-off between super-resolved image quality and time costs than that of the learning-based SISR approaches.

## REFERENCES

- [1] B. Tian *et al.*, “Mapping thermokarst lakes on the Qinghai–Tibet Plateau using nonlocal active contours in Chinese GaoFen-2 multispectral imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1687–1700, May 2017.
- [2] R. Fernandez-Beltran, P. Latorre-Carmona, and F. Pla, “Single-frame super-resolution in remote sensing: A practical overview,” *Int. J. Remote Sens.*, vol. 38, no. 1, pp. 314–354, Jan. 2017.
- [3] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, “SOD-MTGAN: Small object detection via multi-task generative adversarial network,” in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 206–221.
- [4] A. J. Tatem, H. G. Lewis, P. M. Atkinson, and M. S. Nixon, “Super-resolution target identification from remotely sensed images using a hopfield neural network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 4, pp. 781–796, Apr. 2001.
- [5] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, “Is image super-resolution helpful for other vision tasks?” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2016, pp. 1–9.
- [6] D. C. Zanotta, M. P. Ferreira, M. Zorte, and Y. Shimabukuro, “A statistical approach for simultaneous segmentation and classification,” in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 4899–4901.
- [7] B. Hou, K. Zhou, and L. Jiao, “Adaptive super-resolution for remote sensing images based on sparse representation with global joint dictionary model,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2312–2327, Apr. 2018.
- [8] C.-Y. Yang, C. Ma, and M.-H. Yang, “Single-image super-resolution: A benchmark,” in *Proc. Eur. Conf. Comput. Vis.*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 372–386.
- [9] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, “Video super-resolution via deep draft-ensemble learning,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 531–539.
- [10] J. Xu, Y. Liang, J. Liu, and Z. Huang, “Multi-frame super-resolution of Gaofen-4 remote sensing images,” *Sensors*, vol. 17, no. 9, p. 2142, Sep. 2017.
- [11] H. Chavez-Roman and V. Ponomaryov, “Super resolution image generation using wavelet domain interpolation with edge extraction via a sparse representation,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1777–1781, Oct. 2014.
- [12] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [13] K. Turkowski, “Filters for common resampling tasks,” in *Graphics Gems*. New York, NY, USA: Academic, 1990, pp. 147–165.
- [14] M. Irani and S. Peleg, “Improving resolution by image registration,” *Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.
- [15] R. R. Schultz and R. L. Stevenson, “Extraction of high-resolution frames from video sequences,” *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 996–1011, Jun. 1996.
- [16] H. Stark and P. Oskoui, “High-resolution image recovery from image-plane arrays, using convex projections,” *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 6, no. 11, pp. 1715–1726, 1989.
- [17] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [18] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Proc. Int. Conf. Curves Surf.* Berlin, Germany: Springer, 2010, pp. 711–730.
- [19] R. Timofte, V. De, and L. V. Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920–1927.

- [20] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
- [21] 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. (CVPR)*, Jun. 2016, pp. 1646–1654.
- [22] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
- [23] P. Wang, H. Zhang, F. Zhou, and Z. Jiang, "Unsupervised remote sensing image super-resolution using cycle CNN," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 3117–3120.
- [24] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3262–3271.
- [25] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, "Accurate blur models vs. image priors in single image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2832–2839.
- [26] V. Syrris, S. Ferri, D. Ehrlich, and M. Pesaresi, "Image enhancement and feature extraction based on low-resolution satellite data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 1986–1995, May 2015.
- [27] F. Li, L. Xin, Y. Guo, D. Gao, X. Kong, and X. Jia, "Super-resolution for GaoFen-4 remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 1, pp. 28–32, Jan. 2018.
- [28] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 449–456.
- [29] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, Sep. 2002.
- [30] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 783–798.
- [31] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 349–356.
- [32] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5197–5206.
- [33] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3118–3126.
- [34] T. Michaeli and M. Irani, "Nonparametric blind super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 945–952.
- [35] J. M. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "A new deep generative network for unsupervised remote sensing single-image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6792–6810, Nov. 2018.
- [36] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.
- [38] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 701–710.
- [39] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 185–200.
- [40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [41] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 136–144.
- [42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
- [43] 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.*, Jun. 2016, pp. 1874–1883.
- [44] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.
- [45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.
- [46] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
- [47] H. Dong *et al.*, "TensorLayer: A versatile library for efficient deep learning development," in *Proc. ACM Multimedia Conf. (MM)*, 2017, p. 1201.
- [48] Y. Luo, L. Zhou, S. Wang, and Z. Wang, "Video satellite imagery super resolution via convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2398–2402, Dec. 2017.
- [49] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.
- [50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [51] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. Summ. 3rd Annu. JPL Airborne Geosci. Workshop*, vol. 1992, pp. 147–149.
- [52] L. Wald, "Quality of high resolution synthesised images: Is there a simple criterion?" in *Proc. 3rd Conf. Fusion Earth Data, Merging Point Meas., Raster Maps Remotely Sensed Images (SEE/URISCA)*, 2000, pp. 99–103.
- [53] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Comput. Vis. Image Understand.*, vol. 158, pp. 1–16, Dec. 2016.



**Haopeng Zhang** (Member, IEEE) received the B.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2008 and 2014, respectively.

He is an Assistant Professor with the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University. His research interests include remote sensing image processing, multiview object recognition, 3-D object recognition and pose estimation, and other related areas in pattern recognition, computer vision, and machine learning.



**Pengrui Wang** (Member, IEEE) received the B.S. degree from Beihang University, Beijing, China, in 2017. He is pursuing the master's degree with the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University.

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



**Zhiguo Jiang** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Beihang University, Beijing, China, in 1987, 1990, and 2005, respectively.

He is a Professor with the School of Astronautics, Beihang University. His research interests include remote sensing image analysis, target detection, tracking and recognition, and medical image processing.

Dr. Jiang serves as a standing member of the Executive Council of China Society of Image and Graphics and also serves as a member of the Executive Council of the Chinese Society of Astronautics. He is an Editor for the *Chinese Journal of Stereology and Image Analysis*.