PSGAN: A Generative Adversarial Network for Remote Sensing Image Pan-Sharpening

Qingjie Liu, *Member, IEEE*, Huanyu Zhou, Qizhi Xu, *Member, IEEE*, Xiangyu Liu, and Yunhong Wang, *Fellow, IEEE*

***Abstract*— This article addresses the problem of remote sensing image pan-sharpening from the perspective of generative adversarial learning. We propose a novel deep neural network-based method named pansharpening GAN (PSGAN). To the best of our knowledge, this is one of the first attempts at producing high-quality pan-sharpened images with generative adversarial networks (GANs). The PSGAN consists of two components: a generative network (i.e., generator) and a discriminative network (i.e., discriminator). The generator is designed to accept panchromatic (PAN) and multispectral (MS) images as inputs and maps them to the desired high-resolution (HR) MS images, and the discriminator implements the adversarial training strategy for generating higher fidelity pan-sharpened images. In this article, we evaluate several architectures and designs, namely, two-stream input, stacking input, batch normalization layer, and attention mechanism to find the optimal solution for pan-sharpening. Extensive experiments on QuickBird, GaoFen-2, and WorldView-2 satellite images demonstrate that the proposed PSGANs not only are effective in generating high-quality HR MS images and superior to state-of-the-art methods but also generalize well to full-scale images.**

***Index Terms*— Convolutional neural network (CNN), deep learning, generative adversarial network (GAN), pan-sharpening, residual learning.**

I. Introduction

RECENTLY, a lot of high-resolution (HR) optical Earth observation satellites, such as QuickBird, GeoEye, WorldView-2, and GaoFen-2, have been launched, providing researchers in the remote sensing community a large amount of data available for various research fields, such as agriculture [1], land surveying [2], and environmental monitoring [3]. To obtain better results, many of these applications require images at the highest resolution both in spatial and spectral domains. However, due to technical limitations [4], satellites usually carry two kinds of optical imaging sensors and acquire images at two different yet complementary modalities: one is an HR panchromatic (PAN) image and another one is a low resolution (LR) multispectral (MS) image. Pan-sharpening (i.e. panchromatic and MS image fusion), which aims at generating high-spatial-resolution MS images by combining spatial information and spectral information of PAN and MS images, offers us a good solution to alleviate this problem.

Pan-sharpening could be beneficial for many practical applications, such as change detection and land cover classification, so it has gained increasing attention within the remote sensing community. Many research efforts have been devoted to developing pan-sharpening algorithms during the last decades [5]–[11]. The most widely used approaches are so-called component substitution (CS) methods, popularized because of their easy implementation and low computation cost in practical applications [11]–[13]. The basic assumption of CS methods is that the geometric detail information of an MS image lies in its structural component that can be obtained by transforming it into a new space. Then, the structural component is substituted or partially substituted by a histogram matched version of PAN to inject the spatial information. Finally, pan-sharpening is achieved after an inverse transformation. The PCA- [14], [15], the IHS- [6], [12], and the Gram–Schmidt (GS) transform [16]-based methods are those of the most widely known CS methods.

Another popular family is multiresolution analysis (MRA)-based methods. It has a well-known French name amélioration de la résolution spatiale par injection de structures (ARSIS) [17], which means enhancement of the spatial resolution by structure injections. The MRA-based methods assume that the missing spatial information in MS can be inferred from the high frequency of the corresponding PAN image. To pan-sharpen an MS image, multiresolution analysis algorithms, such as discrete wavelet transform (DWT) [18], "à trous" wavelet transform [19], or curvelet transform [20], are applied on a PAN image to extract high-frequency information and then inject it into the corresponding MS image.

In addition, pan-sharpening can be formulated as an inverse problem, in which PAN and MS images are considered as degraded versions of an HR MS image, and it can be restored by resorting to some optimization procedures [21]–[23]. This is an ill-posed problem because much information has been lost during the degrading process. To obtain the optimal solution, regularizer [21] or prior knowledge [22] is added into formulations, or pan-sharpening can be addressed from the perspective of machine learning. For instance, Li *et al.* [24]

Manuscript received March 20, 2020; revised June 16, 2020 and November 9, 2020; accepted November 29, 2020. Date of publication December 24, 2020; date of current version November 24, 2021. This work was supported by NSFC under Grant 41871283 and Grant 61601011. *(Corresponding author: Yunhong Wang.)*

Qingjie Liu, Huanyu Zhou, and Yunhong Wang are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Hangzhou 310051, China (e-mail: qingjie.liu@buaa.edu.cn; zhysora@buaa.edu.cn; yhwang@buaa.edu.cn).

Qizhi Xu is with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China (e-mail: qizhi@buaa.edu.cn).

Xiangyu Liu was with the School of the Computer Science and Engineering, Beihang University, Beijing 100191, China.

Digital Object Identifier 10.1109/TGRS.2020.3042974

and Zhu *et al.* [25], [26] modeled pan-sharpening from compressed sensing theory. Liu *et al.* [27] addressed pan-sharpening from a manifold learning framework.

Recently, deep learning techniques have achieved great success in diverse computer vision tasks [28]–[32], inspiring us to design deep learning models for the pan-sharpening problem. Observing that pan-sharpening and single-image super-resolution share a similar spirit and motivated by [28], Masi *et al.* [9] proposed a three-layered convolutional neural network (CNN)-based pan-sharpening method and obtained improved results than traditional algorithms, such as BDSD [33] and AWLP [34]. Following this work, increasing attention has been paid to deep learning-based pan-sharpening. For instance, Zhong *et al.* [35] presented a CNN-based hybrid pan-sharpening method. Different from [9] that generates the pan-sharpened MS images directly, Zhong *et al.*'s work first enhances the spatial resolution of an input MS with the SRCNN method [28] and then applies GS transform on the enhanced MS and the PAN to accomplish the pan-sharpening. Rao *et al.* [36] proposed a CNN-based pan-sharpening model built on top of SRCNN, in which SRCNN was employed to learn the difference between upsampled MS image and ground truth. The final results were obtained by adding the predicted difference image to the upsampled MS. Similarly, Wei *et al.* [37] proposed a much deeper network (11 layers) to learn the residual images.

Recent studies [38]–[40] have suggested that deeper networks will achieve better performance on vision tasks. However, training becomes very difficult with depth increasing. Residual learning [41] eases this problem by introducing shortcut connections between different layers of a network, allowing training networks much deeper than previous ones. Pan-sharpening could also be improved by residual learning. Although Rao *et al.* [36] and Wei *et al.* [37] used the concept "residual network," the networks employed in their methods are built with plain units. The depth of their networks is still shallow. The first attempt at applying the residual network is PanNet [10]. They adopt a similar idea to [36] and [37] but employ ResNet [41] to predict details of the image. In this way, both spatial information and spectral information could be preserved well.

Although great advances have been made in this field, there is still a great gap between the synthetic HR MS and the real one. It is still a challenging problem for researchers in the remote sensing community to obtain high spectral and spatial fidelity pan-sharpened images. To further boost the performance of pan-sharpening networks and obtain high-quality pan-sharpened images, in this article, we reformulate pan-sharpening as an image generation problem and explore the utilization of generative adversarial network (GAN) [42], [43] to solve it. The GAN framework is a powerful generative model and was first introduced by Goodfellow *et al.* [42]. In contrast to previous networks that have a unified architecture, GANs have two individual components: one generator that is trained to generate images indistinguishable from real ones and one discriminator that tries to distinguish whether the generated images are real or fake. With this perspective, this article proposes pansharpening GAN (PSGAN), a GAN that could produce high-quality pan-sharpened images conditioned on the input of PAN and LR MS images.

This is an extension of our previous work [44], which is the first work that addresses the pan-sharpening problem from the perspective of generative adversarial learning. Compared with [44], background knowledge about GAN is presented. We give more details about the architecture of the proposed PSGAN and evaluate several possible architecture configurations of the PSGAN. We enlarge the data set and conduct extensive experiments to demonstrate the effectiveness and superiority of it. The main contributions of this article are as follows.

1) We address the pan-sharpening problem from the perspective of image generation and develop novel GANs for solving it.
2) To accomplish pan-sharpening with the GAN framework, we design a basic two-stream CNN architecture as the generator to produce high-quality pan-sharpened images and employ a fully convolutional discriminator to learn adaptive loss function for improving the quality of the pan-sharpened images.
3) We evaluate various configurations of the proposed PSGAN and distill important factors that affect the performance of the pan-sharpening task.
4) We demonstrate that the proposed PSGAN can produce astounding results on the pan-sharpening problem. Fig. 1 shows one example result produced by our method. Codes are available.[1]

The remainder of this article is organized as follows. Backgrounds and the theory of GANs are briefly introduced in Section II. Section III formulates pan-sharpening from the perspective of generative adversarial learning and gives details of proposed PSGAN architecture. Experiments are conducted in Section IV. Finally, this article is concluded in Section V.

II. Generative Adversarial Networks

Given a set of unlabeled data, generative models aim at estimating their underlying distributions. This is a highly challenging task, and inference on such distributions could be computationally expensive or even intractable. Recently proposed GANs (GANs) [42] provide an efficient framework to learn generative models from the unlabeled data.

GANs learn generative models by setting up an adversarial game between a generator neural network G and a discriminator neural network D. For any given data set $\{\mathbf{x}\}$, the generator G learns the distribution of the data by mapping a random sample $\mathbf{z}$ from any distributions (e.g., the Gaussian distribution or uniform distribution) to a sample $\mathbf{x}$ from the data space. The G is trained to produce samples that cannot be distinguished from the real samples. The discriminator D outputs a scalar indicating the probability that the samples are produced by G, or it is from the real distribution. This process can be formulated as a two-player min–max game and written

[1] https://github.com/zhysora/PSGan-Family

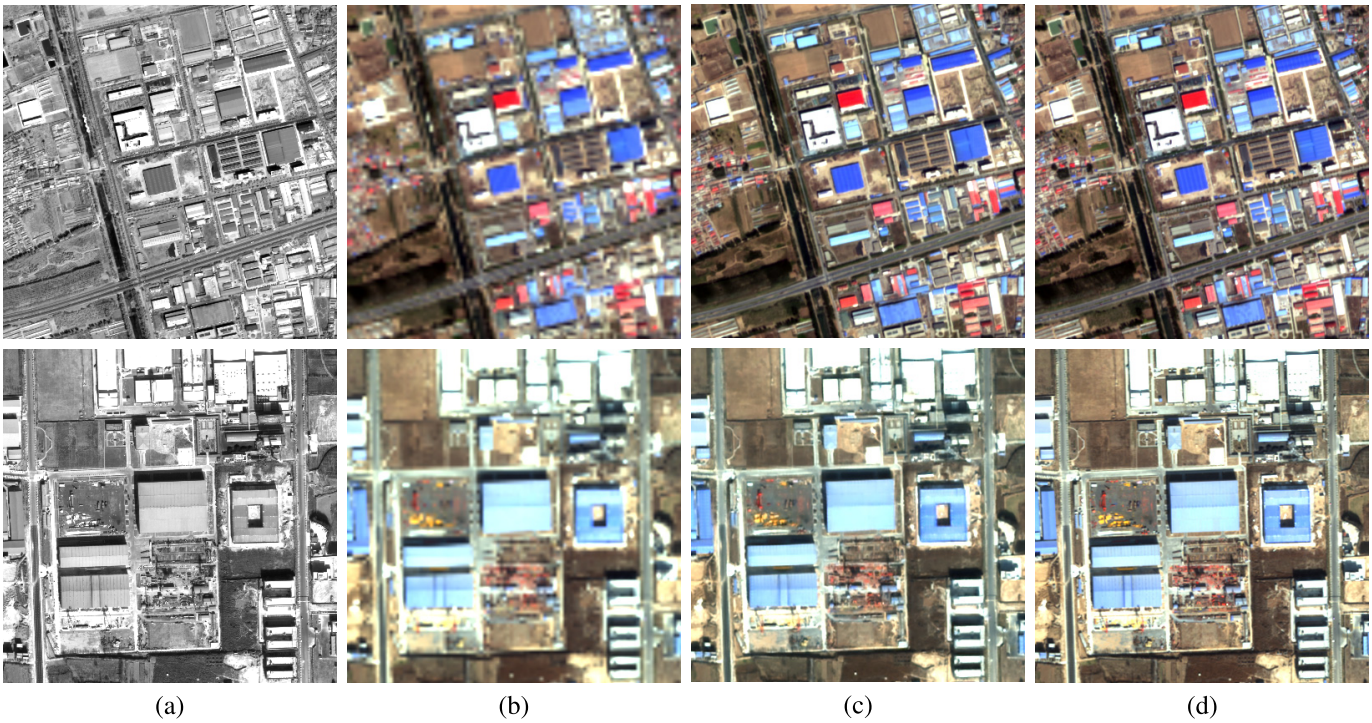

Fig. 1. Example results of our PSGAN method. (c) Desired HR MS images generated from (a) PAN and (b) LR MS. (d) Ground-truth HR MS images.

as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \quad (1)$$

where $p_{\text{data}}(\mathbf{x})$ is the distribution of the real data, and $\mathbf{x}$ is a sample from $p_{\text{data}}(\mathbf{x})$. Correspondingly, $p_z(\mathbf{z})$ is an arbitrary random distribution, and $\mathbf{z}$ is a sample drawn from it. The first term in the right side of (1) indicates the probability of the discriminator determining a sample being a "real" data, while the second term indicates the probability of the discriminator identifying a sample being "fake." D tries to assign correct labels to both real and generated data by maximizing the first term to 1 and the second term to 0. In contrast, G takes a random noise $\mathbf{z}$ as input and tries to generate a sample that as indistinguishable from the real one as possible by minimizing $\log(1 - D(G(\mathbf{z})))$. An illustration of this procedure is given in Fig. 2.

Equation (1) can be optimized in an iterative way by fixing one parameter and optimizing another one. When G is fixed, the optimization of D can be considered as maximizing the log-likelihood of the conditional probability $p(Y = y|\mathbf{x})$, where Y is the probability that the sample $\mathbf{x}$ comes from the real data ($y = 1$) or the fake data ($y = 0$). When D is fixed, the objective of G is minimizing the Jensen–Shannon divergence between the real data distribution p_{data} and the fake data distribution p_G (here, p_G denotes distribution learned by the generator G). It can be proved that G has an optimal solution $p_G = p_{\text{data}}$ [42]. Given enough capacity and training time, the generative neural network and the discriminator network will converge and achieve a point where the generator produces samples so real that the discriminator cannot distinguish them from the real data.

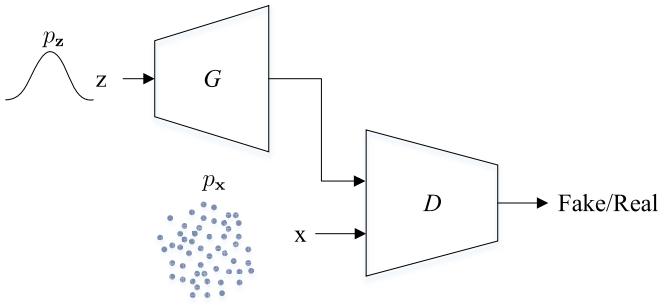

Fig. 2. Illustration of the generative adversarial framework. G is a generator accepting a random signal $\mathbf{z}$ and trained to generate output that cannot be distinguished from a real data $\mathbf{x}$ by a discriminator D.

III. PSGAN

A. Formulation

Pan-sharpening aims to estimate a pan-sharpened HR MS image $\hat{\mathbf{P}}$ from an LR MS image $\mathbf{X}$ and an HR PAN image $\mathbf{Y}$. The output images should be as close as possible to the ideal HR MS images $\mathbf{P}$. We describe $\mathbf{X}$ by a real-valued tensor of size $w \times h \times b$, $\mathbf{Y}$ by $rw \times rh \times 1$, and $\hat{\mathbf{P}}$ and $\mathbf{P}$ by $rw \times rh \times b$, respectively, where r is the spatial resolution ratio between LR MS $\mathbf{X}$ and HR PAN $\mathbf{Y}$ (in this article, $r = 4$) and b is the number of bands. The ultimate goal of pan-sharpening takes a general form as follows:

$$\hat{\mathbf{P}} = f(\mathbf{X}, \mathbf{Y}; \Theta) \quad (2)$$

where $f(\cdot)$ is a pan-sharpening model that takes $\mathbf{X}$ and $\mathbf{Y}$ as input and produces THE desired HR MS $\hat{\mathbf{P}}$, and Θ is THE collection of parameters for this model. Equation (2) can be

solved by minimizing the following loss function:

$$\hat{\Theta}_f = \arg\min \sum_{n=1}^{N} \ell[f_\Theta(\mathbf{X}_n, \mathbf{Y}_n), \mathbf{P}_n] \tag{3}$$

where N is the number of training samples. As an example, (2) can be realized from the perspective of compressed sensing, and (3) can be solved using dictionary learning algorithms [45].

From (2), we can see that $f(\cdot)$ can be considered as a mapping function from $(\mathbf{X}, \mathbf{Y})$ to $\mathbf{P}$. Thus, we can reformulate pan-sharpening as a conditional image generation problem that can be solved using conditional GAN [46]. Following [42] and [46], we define a generative network G that maps the joint distribution $p_{\text{data}}(\mathbf{X}, \mathbf{Y})$ to the target distribution $p_r(\mathbf{P})$. The generator G tries to produce pan-sharpened image $\hat{\mathbf{P}}$ that cannot be distinguished from the reference image $\mathbf{P}$ by an adversarial trained discriminative network D. This can be expressed as a min–max game problem

$$\min_{\Theta_G} \max_{\Theta_D} \mathbb{E}_{\mathbf{X}\sim p_{data}(\mathbf{X}),\mathbf{P}\sim p_r(\mathbf{P})}[\log D_{\Theta_D}(\mathbf{X}, \mathbf{P})] + \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p_{data}(\mathbf{X},\mathbf{Y})}[\log(1 - D_{\Theta_D}(\mathbf{X}, G_{\Theta_G}(\mathbf{X}, \mathbf{Y})))]. \tag{4}$$

With this adversarial learning, a GAN designed for pan-sharpening tasks could generate faithful HR MS images.

B. Architectures of the Generator

The ultimate goal of a generator is to produce a pan-sharpened MS image that cannot be distinguished from a real MS image. Since the inputs for a generator G are an HR PAN image and an LR MS image, there are multiple ways to design the G. One possible way is using the network architecture similar to PNN [9] that stacks the PAN and the upsampled MS to form a five-band input.[2] Another way is directly taking a two-stream design as in [47]. In this work, we devise and evaluate several generator architectures.

1) Two-Stream Generator: In contrast to other image generation tasks, e.g. single-image super-resolution [31], image dehazing [48], or face aging [49], that learn one-to-one mappings, pan-sharpening accepts two images acquired by different sensors with distinct characteristics over the same scene. The two modalities, i.e., the PAN image and the MS image, contain different information. PAN image is the carrier of geometric detail (spatial) information, while MS image preserves spectral information. To make the best use of spatial information and spectral information, we utilize two subnetworks to extract the hierarchical features of the input PAN and MS to capture complementary information of them. After that, the subsequent network proceeds as an autoencoder: the encoder fuses information extracted from PAN and MS images, and the decoder reconstructs the HR MS images from the fused features in the final part.

Considering that the spatial resolution of MS images is only $1/4$ of the desired pan-sharpened MS images, the pan-sharpening can be viewed as a special case of image super-resolution aided by the PAN image. To enhance the spatial resolution of MS images using neural networks, there are usually two solutions. The first one is upscaling MS images to the desired size using some interpolation methods, such as bicubic, and then applying neural networks to learn nonlinear mapping. The second one is applying the model directly without any preprocessing and performing upscaling using networks. This will lead to deeper network structure and potentially better performance, however, with lower computational cost than the previous one [50]. In this article, we take into consideration both of the two solutions.

PSGAN: In our previous work [44], we employ the first solution to build the PSGAN, which is upsampling the MS image first and then feeding it and the corresponding PAN into two subnetworks for feature extraction. The architecture of the generator is shown in Fig. 3. The two subnetworks have a similar structure but different weights. Each of them consists of two successive convolutional layers followed by a leaky rectified linear unit (LeakyReLU) [51] and a downsampling layer. The convolutional layer with a stride of 2 instead of a simple pooling strategy, e.g., max pooling, is used to downsample the feature maps. After passing through the two subnetworks, the feature maps are first concatenated and then fused by subsequent convolutional layers. Finally, a decoder-like network architecture comprised of two transposed convolutional and three-flat convolutional layers is applied to reconstruct the desired HR MS images. Inspired by the U-Net [52], we adapt the PSGAN network by adding skip connections. The skip connection will not only compensate details to higher layers but also ease the training. In the last layer, ReLU is used to guarantee the output is not negative.

FU-PSGAN: We take PSGAN as the base network and build variations on top of it. To differentiate different versions of PSGAN, we name the PSGAN with the second solution FU-PSGAN because the generator of it uses the **F**eature **U**p-scaling strategy. The generator of FU-PSGAN has almost identical architecture to PSGAN except that the MS subnetwork takes the original-sized MS as input and has one up convolution following the first convolution layer instead of a normal convolution layer, as shown in Fig. 4(a).

2) PAN and MS Stacked Generator: Another possible way of designing generators is viewing the PAN and MS as a whole, i.e. stacking the two images along the channel dimension together to form a new image. To do this, the MS image should be upsampled to match the size of the PAN image and then concatenated with the PAN to obtain an inflated image. One appealing advantage of this strategy is we can easily inherit some well-developed models from related research fields, such as single-image super-resolution. For example, the pioneering PNN [9] borrows the main structure of the network from SRCNN [28].

ST-PSGAN: Following PNN [9], we design a deeper residual network to accomplish pan-sharpening. We call it ST-PSGAN since it has a **ST**acked generator. The generator of ST-PSGAN is shown in Fig. 4(b). It has almost the same structure as the generator of PSGAN in Fig. 3 except for one major difference that one stream, along with the skip connection bound on it, is removed to be consistent with the stacked PAN and MS. Another imperceptible change is that the first

[2]In this article, we only consider four-band MS images.

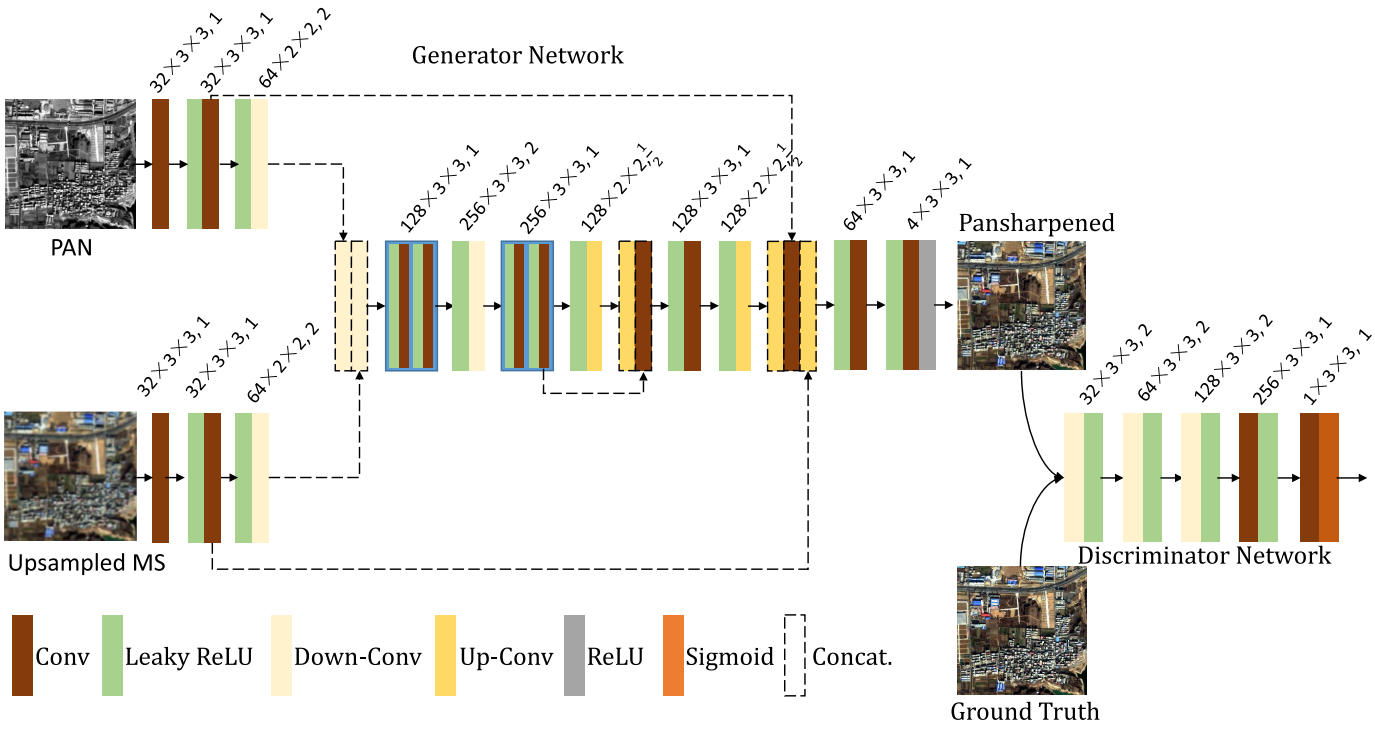

Fig. 3. Detailed architectures of the generator network G and the discriminator network D.

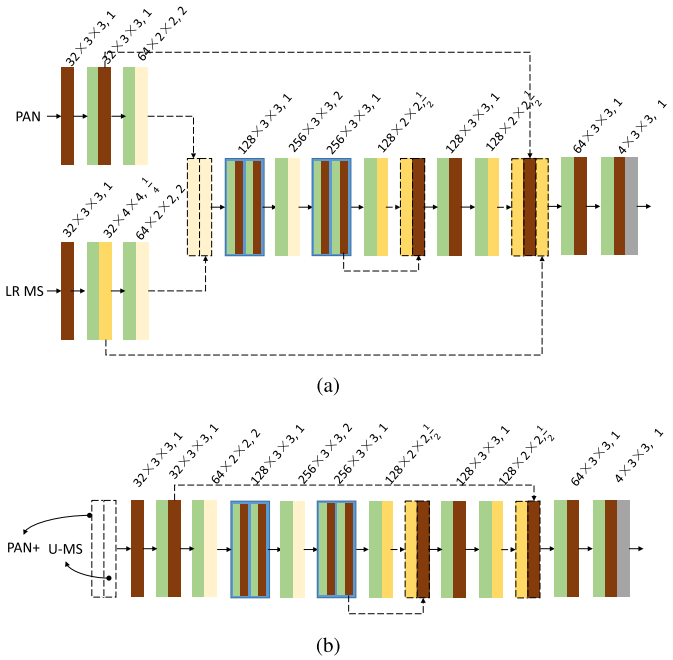

Fig. 4. Two variant generators for PSGANs. (a) Generator performing upscaling using networks. (b) Stacked generator that accepts one concatenated PAN and Upsampled MS (U-MS) image as input. Note: (a) and (b) share the same legend as Fig. 3.

convolution layer should adapt to the channel dimension of the new input. For fair comparisons, all three PSGANs share the same discriminator, which will be described in the following.

C. Fully Convolutional Discriminator

In addition to the generator, a conditional discriminator network is trained simultaneously to discriminate the reference MS images from the generated pan-sharpened images. Similar to [46], we use a fully convolutional discriminator that consists of five layers with kernels of 3×3. The stride of the first three layers is set to 2, and 1 for the last two layers. Except for the last layer, all the convolution layers are activated through LeakyReLU. Sigmoid is used to predict the probability of being real HR MS or pan-sharpened MS for each input. The architecture of the discriminator is shown in Fig. 3.

We give the detailed parameters of the proposed PSGANs in Figs. 3 and 4. Taking into account the tradeoff between the model complexity and the performance, we do not build much deeper networks although the depths of them can be deepened easily by inserting more convolution blocks. Also, larger kernels, such as 5×5 or 7×7, are not considered in this work because they bring much more parameters with the same network depth.

D. Loss Function

We train the three models using the same loss function. In this section, we will take PSGAN as an example to describe the loss function. The generative network G and the discriminator network D are trained alternately. To optimize G, we adopt the pixelwise loss and adversarial loss similar to some other state-of-the-art GAN networks [46]. In contrast to many previous works [9], [10] employing ℓ_2 loss that calculates mean squared errors between the ground truth and the reconstructed images, in this work, we adopt ℓ_1 loss that calculates the absolute difference between the pan-sharpened image and the ground truth

$$\mathcal{L}(G) = \sum_{n=1}^{N} [-\alpha \log D_{\Theta_D}(\mathbf{X}, G_{\Theta_G}(\mathbf{X}, \mathbf{Y})) + \beta \|\mathbf{P} - G_{\Theta_G}(\mathbf{X}, \mathbf{Y})\|_1]. \quad (5)$$

TABLE I
BRIEF INFORMATION ABOUT THE THREE DATA SETS USED IN EXPERIMENTS. NOTE THAT SPA. RES. MEANS SPATIAL RESOLUTION

Dataset	Images (Train/Test)	Training Samples	Spa. Res. (PAN/MS)
QB	9 (8/1)	25,038	0.6/2.4
GF-2	9 (8/1)	13,460	0.8/3.2
WV-2	9 (8/1)	11,552	0.5/2.0

Finally, the loss function for D takes the form

$$\mathcal{L}(D) = \sum_{n=1}^{N}[1 - \log D_{\Theta_D}(\mathbf{X}, G_{\theta_G}(\mathbf{X}, \mathbf{Y})) + \log D_{\Theta_D}(\mathbf{X}, \mathbf{P})] \tag{6}$$

where N is the number of training samples in a minibatch, and α and β are the hyperparameters and are set to 1 and 100 in the experiments, respectively.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the effectiveness and superiority of the proposed PSGANs.

A. Data Set and Implementation Details

We train and test our networks on three data sets comprised of images acquired by QuickBird (QB), GaoFen-2 (GF-2), and WorldView-2 (WV-2) satellite images. Since the desired HR MS images are not available, we follow Wald's protocol [53] to downsample both the MS and PAN images with a factor of r ($r = 4$ in this article). Then, the original MS images are used as reference images to be compared with. We randomly crop patch pairs from the downsampled MS and PAN to form training samples. It should be noted that we use larger patch size, $64 \times 64 \times 4$ for MS patches and $256 \times 256 \times 1$ for PAN patches, than our previous work [44], in which the sizes for MS and PAN patches are $32 \times 32 \times 4$ and $128 \times 128 \times 1$, respectively. This will lead to a smaller batch size during training; however, our experiments, which will be given in the next subsection, demonstrate that larger patch size produces better image quality. Brief information about the three data sets is illustrated in Table I. All the results reported in the following sections are based on the test sets which are independent of the training images.

The PSGANs are implemented in PyTorch [54] and trained on a single NVIDIA Titan 2080Ti GPU. We use Adam optimizer [55] with an initial learning rate of 0.0002 and a momentum of 0.5 to minimize the loss function. The minibatch size is set to 8. It takes about 8 h to train one model. The source codes and more experimental results are available at https://github.com/zhysora/PSGan-Family.

B. Evaluation Indexes

We use five widely used metrics to evaluate the performance of the proposed and other methods on the four data sets, including SAM [56], CC, sCC [57], ERGAS [53], and Q_4 [58].

1) *SAM:* The spectral angle mapper (SAM) [56] measures spectral distortions of pan-sharpened images comparing with the reference images. It is defined as angles between the spectral vectors of pan-sharpened and reference images in the same pixel, which can be calculated as

$$\mathrm{SAM}(\mathrm{x}_1, \mathrm{x}_2) \triangleq \arccos\left(\frac{\mathrm{x}_1 \cdot \mathrm{x}_2}{\| \mathrm{x}_1 \| \cdot \| \mathrm{x}_2 \|}\right) \tag{7}$$

where x_1 and x_2 are two spectral vectors. SAM is averaged over all the images to generate a global measurement of spectral distortion. For the ideal pan-sharpened images, SAM should be 0.

2) *CC:* The correlation coefficient (CC) is another widely used indicator measuring the spectral quality of pan-sharpened images. It calculates the CC between a pan-sharpened image X and the corresponding reference image Y as

$$\mathrm{CC} \triangleq \frac{\sum_{i=1}^{w}\sum_{j=1}^{h}\left(X_{i,j} - \mu_X\right)\left(Y_{i,j} - \mu_Y\right)}{\sqrt{\sum_{i=1}^{w}\sum_{j=1}^{h}\left(X_{i,j} - \mu_X\right)^2 \sum_{i=1}^{w}\sum_{j=1}^{h}\left(Y_{i,j} - \mu_Y\right)^2}} \tag{8}$$

where w and h are the width and height of the images, and μ_* indicates the mean value of an image. CC ranges from -1 to $+1$, and the ideal value is $+1$.

3) *sCC:* To evaluate the similarity between the spatial details of pan-sharpened images and reference images, a high-pass filter is applied to obtain the high frequencies of them, and then, the CC between the high frequencies is calculated. This quantity index is called spatial CC (sCC) [57]. We use the high Laplacian pass filter given by

$$F = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{9}$$

to get a high frequency. A higher sCC indicates that most of the spatial information of the PAN image is injected during the fusion process. sCC is computed between each band of the pan-sharpened and reference image. The final sCC is averaged over all the bands of the MS images.

4) *ERGAS:* The erreur relative globale adimensionnelle de synthèse (ERGAS), also known as the relative global dimensional synthesis error, is a commonly used global quality index [53]. It is given by

$$\mathrm{ERGAS} \triangleq 100\frac{h}{l}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathrm{RMSE}(B_i)}{M(B_i)}\right)^2} \tag{10}$$

where h and l are the spatial resolution of PAN and MS images; $\mathrm{RMSE}(B_i)$ is the root mean square error between the ith band of the fused and reference image; $M(B_i)$ is the mean value of the original MS band B_i.

TABLE II
PERFORMANCE VARIATIONS WITH RESPECT TO PATCH SIZES. THE BATCH SIZE DECREASES FROM 64 TO 8 WITH PATCH SIZE INCREASING FROM 16 TO 64

		Patch Size	SAM ↓	CC ↑	sCC ↑	ERGAS ↓	Q4 ↑
QB	PSGAN	16	1.2514	0.9862	0.9867	1.3703	0.9852
		32	1.2270	0.9867	0.9869	1.3594	0.9852
		64	1.1740	0.9877	0.9880	1.2602	0.9869
	FU-PSGAN	16	1.3338	0.9855	0.9853	1.4167	0.9842
		32	1.2883	0.9857	0.9861	1.3943	0.9845
		64	1.2411	0.9869	0.9865	1.2907	0.9864
	ST-PSGAN	16	1.3114	0.9874	0.9864	1.3225	0.9865
		32	1.3125	0.9867	0.9869	1.3662	0.9854
		64	1.2889	0.9869	0.9868	1.3267	0.9857
GF-2	PSGAN	16	0.7632	0.9905	0.9927	0.7388	0.9981
		32	0.7484	0.9908	0.9929	0.7303	0.9981
		64	0.7575	0.9909	0.9929	0.7233	0.9980
	FU-PSGAN	16	0.7527	0.9908	0.9930	0.7287	0.9981
		32	0.7456	0.9909	0.9931	0.7214	0.9982
		64	0.7181	0.9915	0.9935	0.7013	0.9982
	ST-PSGAN	16	0.8325	0.9894	0.9914	0.7832	0.9978
		32	0.7856	0.9903	0.9924	0.7477	0.9980
		64	0.7300	0.9913	0.9933	0.7084	0.9981

5) Q_4: The quality index Q_4 [58] is the four-band extension of Q index [59]. Q_4 is defined as

$$Q_4 \triangleq \frac{4|\sigma_{z_1z_2}| \cdot |\mu_{z_1}| \cdot |\mu_{z_2}|}{(\sigma_{z_1}^2 + \sigma_{z_2}^2)(\mu_{z_1}^2 + \mu_{z_2}^2)} \tag{11}$$

where z_1 and z_2 are two quaternions, formed with spectral vectors of MS images, i.e., $z = a + \mathbf{i}b + \mathbf{j}c + \mathbf{k}d$, μ_{z_1} and μ_{z_2} are the means of z_1 and z_2, $\sigma_{z_1z_2}$ denotes the covariance between z_1 and z_2, and $\sigma_{z_1}^2$ and $\sigma_{z_2}^2$ are the variances of z_1 and z_2.

Three nonreference metrics, D_λ, D_S, and QNR, are employed for full-resolution assessment.

1) D_λ [60] is a spectral quality indicator derived from the difference of interband Q values calculated from the pan-sharpened MS bands and the low-resolution MS bands. It is defined as

$$D_\lambda \triangleq \sqrt{\frac{2}{K(K-1)} \sum_{i=1}^{K} \sum_{j=i}^{K} \left| Q(P_i, P_j) - Q(X_i, X_j) \right|} \tag{12}$$

where K is the number of bands for a MS image, and P_i and X_i represent the ith band of the pan-sharpened and the LR MS images, respectively.

2) D_S [60] is a spatial quality metric complementary to D_λ. It is calculated as

$$D_S \triangleq \sqrt{\frac{1}{K} \sum_{i=1}^{K} \left| Q(P_i, Y) - Q(X_i, \tilde{Y}) \right|} \tag{13}$$

where Y is a PAN image and $\tilde{Y}$ is its degraded low-resolution version. Both D_λ and D_S take values in [0,1], and the lower the better.

3) *QNR:* [60] is the abbreviation of Quality with No Reference. It is a combination of D_λ and D_S and measures global quality of fused images without any reference image. It is given by

$$\text{QNR} \triangleq (1 - D_\lambda)(1 - D_S). \tag{14}$$

The ideal value of QNR is 1.

C. Impact of Patch Size

In our previous work [44], we use small patch size to generate training samples, which allows us to set a larger batch size and, thus, enables more stable training and faster convergence [61]. However, for image reconstruction tasks, larger patch size is beneficial for generating high-quality images. In this article, we test a much larger patch size than [44]. Although the batch size will decrease accordingly, our experiments demonstrate that larger patches lead to higher image quality. We conduct experiments on the QB and GF-2 images to evaluate how much the impact will be by setting different patch sizes. The results are given in Table II, from which we can see larger patch size does have a positive impact on the image quality. For all the three models, the image quality has a significant improvement when using the 64 patch size even though the batch is decreased from 32 to 8. Especially, the spectral indicator SAM and the global quality measurement ERGAS have obvious superiority with one exception that the SAM for PSGAN has slightly lower value on GF-2 images. The effect of patch size on CC, sCC, and Q4 is weak, only with small improvements and, sometimes, even slightly worse, even though we are encouraged to use larger patch size since the spectral quality is very important in the pan-sharpening task. Thus, in the following experiments, the patch size is set as 64.

D. Impact of Number of Feature Maps and Kernel Size

The kernel size and the number of feature maps are important factors when designing neural networks. We test two designs of our PSGAN to evaluate the impacts of feature maps and kernel sizes. First, we reduce the number of feature maps by half, thus leading to fewer parameters. Second, we replace the typical 3×3 convolutional filters with 5×5 kernels. Enlarging kernel size would increase the number of parameters of the model, dramatically, as shown in Table VII. We name these two designs PSGAN-f16 and PSGAN-k5 $\times$ 5, respectively. Test results are given in Table III. Although PSGAN-f16 has fewer parameters (about 1/4 of PSGAN) and runs

TABLE III
IMPACTS OF BN AND SA ON THE FOUR DATA SETS. "+BN" MEANS THAT PSGAN IS MODIFIED BY ADDING BATCH NORMALIZATION AFTER EACH CONVOLUTION BLOCK. "+SA" MEANS THAT PSGAN IS EQUIPPED WITH SA

		SAM↓	CC↑	sCC↑	ERGAS↓	Q_4↑
QB	PSGAN	1.1740	0.9877	0.9880	1.2602	0.9869
	PSGAN-f16	1.5760	0.9832	0.9818	1.4414	0.9832
	PSGAN-k5 × 5	1.2843	0.9845	0.9863	1.4316	0.9826
	PSGAN+BN	9.2521	0.9274	0.9075	14.029	0.8167
	PSGAN+SA	2.1396	0.9696	0.9738	2.3140	0.9629
GF-2	PSGAN	0.7575	0.9909	0.9929	0.7233	0.9980
	PSGAN-f16	0.8411	0.9889	0.9908	0.7994	0.9976
	PSGAN-k5 × 5	0.7293	0.9914	0.9932	0.7025	0.9981
	PSGAN+BN	1.5903	0.9669	0.9698	7.6059	0.9260
	PSGAN+SA	1.2182	0.9739	0.9793	1.2422	0.9944
WV-2	PSGAN	0.9127	0.9973	0.9975	1.6452	0.9971
	PSGAN-f16	0.9955	0.9968	0.9970	1.7718	0.9966
	PSGAN-k5 × 5	1.0061	0.9972	0.9971	1.7002	0.9968
	PSGAN+BN	8.1520	0.8697	0.8448	9.6634	0.8016
	PSGAN+SA	1.4185	0.9950	0.9943	2.2273	0.9944

faster than other PSGANs, its performance is not satisfactory. It obtains better results than PSGAN+BN and PSGAN+SA but weaker than PSGAN and PSGAN-k5 × 5. PSGAN-k5 × 5 is with more than 2× parameters than PSGAN and about 10× parameters than PSGAN-f16. Such huge parameters make networks hard to train. From Table III, we can see that PSGAN-k5 × 5 works well even better than PSGAN on the GF-2 images and, however, obtains worse results on the QB and WV-2 data sets.

E. Batch Normalization is Harmful

Batch normalization (BN) [62] has been widely used in neural networks to stabilize and accelerate training. It also has been applied to the pan-sharpening task for improving performance [63], [64]. However, recent studies have suggested that BN may be unnecessary in low-level visions [65]. It brings two burdens. First, BN operation requires an amount of storage and computational resources, which could be used to add more convolutional layers. Second, BN layers get rid of scale information, which is helpful for recognition tasks and, however, is harmful to scale-sensitive tasks, such as image super-resolution and pan-sharpening. We add a BN layer into each block for comparison. The results are given in Table III. We can observe that adding BN layers severely decreases the performance, especially on the QB and WV-2 images. Thus, in this work, we remove all the BN layers from our models.

F. Self-Attention is Not Useful

Attention mechanism plays an important role in human perception. It allows human brains to selectively concentrate on information meaningful to perceive tasks while ignoring other irrelative information. Since it was introduced to deep learning [73], the attention mechanism has become one of the most valuable breakthroughs in the community and significantly boosts a variety of AI tasks ranging from NLP [74] to CV [75] domains.

Among many attention models, self-attention (SA) has been reported to be able to generate high-quality images when incorporating GANs [75]. Thus, in this article, we explore to leverage the SA module to improve PSGAN. Following [75], the nonlocal model [76] is adopted to introduce SA to our PSGAN. To be specific, the SA module is added into the ninth layer of the generator and the last layer of the discriminator. The experimental results are illustrated in Table III, from which we can see that, on the QB and WV-2 images, PSGAN+SA performs much better than PSGAN+BN and, however, still worse than the original PSGAN. Although the CC and sCC on WV-2 images and the Q_4 on GF-2 and WV-2 images are satisfactory to an extent, SA is not welcome in our models.

G. Two-Stream is Better Than Stacking

We present two variants for our PSGAN, i.e., FU-PSGAN and ST-PSGAN. PSGAN and FU-PSGAN share a similar structure that both have two-stream inputs, as described in Section III-B1. ST-PSGAN has only one input branch that accepts stacked PAN and upsampled MS as input. Most previous works adopt a one branch design similar to ST-PSGAN, such as PNN [9] and PanNet [10], and ignore the two-stream solution. To achieve a better performance, we evaluate different structures of PSGAN and report quantitative results in Tables IV–VI with all metrics, including nonreference ones that are given. It should be noted that all nonreference measurements are calculated under the full-scale image setting. From these tables, we can observe that the stacking strategy, i.e. ST-PSGAN, is the worst among the three PSGANs in almost all cases except for on the GF-2 images (see Table V) where it obtains the second best results. Generally speaking, the two-stream strategy is better than stacking, and the models built with the two-stream idea are expected to achieve better performances. FU-PSGAN reaches the top performance on GF-2 images (see Table V) and obtains satisfactory results on WV-2 (see Table VI). On the QB set, it is inferior to PSGAN and, however, still better than ST-PSGAN in terms of all metrics except for sCC. The three PSGANs generalize well to full-scale images. Although ST-PSGAN achieves the best D_λ and QNR on GF-2 images, it still lags behind the other two PSGANs on QB and WV-2 images. FU-PSGAN performs the best on WV-2 and, however, the worst on GF-2 images. PSGAN is superior to the other variants on QB with the lowest D_λ and D_S and the highest QNR.

Although the quantitative measures vary in terms of numerical metrics, the visual perceptions of them are very similar, as shown in Figs. 5(m)–(o), and 6(m)–(o), and 7(f)–(h). In Figs. 5(m)–(o) and 6(m)–(o), all of them have faithful colors and spatial details to the ground-truth images. Fig. 7(f)–(h) shows the results on full-scale images. Careful inspection of them indicates that FU-PSGAN is the best among the three on WV-2 images, which is consistent with the quantitative results in Table VI.

H. Comparison With Other Pan-Sharpening Methods

In this section, we compare the proposed PSGAN and its two improved variations, i.e. FU-PSGAN and ST-PSGAN with 12 widely used pan-sharpening techniques, including ten traditional methods: SFIM [66], LMVM [67], LMM [67],

TABLE IV

PERFORMANCE COMPARISONS ON THE TEST SET OF QB. THE TOP-THREE PERFORMANCES ARE HIGHLIGHTED WITH **RED**, **GREEN**, AND **BLUE**

	SAM↓	CC↑	sCC↑	ERGAS↓	Q4↑	$D_\lambda \downarrow$	$D_S \downarrow$	QNR↑
SFIM [66]	1.3465	0.9620	0.9752	2.6051	0.9643	**0.0062**	0.0170	0.9769
LMVM [67]	1.7131	0.9694	0.9703	2.3509	0.9647	**0.0020**	0.0164	**0.9816**
LMM [67]	1.6845	0.9634	0.9695	2.4306	0.9640	0.0064	0.0173	0.9763
HPF [68]	1.3522	0.9699	0.9811	2.2534	0.9698	0.0069	0.0178	0.9755
HPFC [68]	1.6558	0.9609	0.9776	4.2814	0.9453	0.0461	0.0468	0.9093
Brovey [69]	1.4782	0.9729	0.9720	2.0542	0.9735	0.0281	0.0503	0.9231
HCS [70]	1.4782	0.9729	0.9685	2.5003	0.9632	0.0137	0.0285	0.9582
IHS [11]	1.6100	0.9683	0.9822	2.2611	0.9697	0.0078	0.0550	0.9376
GS [71]	1.3063	0.9726	0.9821	2.1309	0.9704	0.0232	0.0497	0.9283
BDSD [33]	1.4725	0.9725	0.9864	2.2722	0.9707	0.0147	0.0227	0.9629
PNN [9]	2.0777	0.9731	0.9718	1.8752	0.9723	0.0273	0.0278	0.9457
PanNet [10]	**1.1068**	**0.9848**	**0.9877**	1.3800	0.9834	**0.0019**	**0.0111**	**0.9871**
RED-cGAN [72]	1.2541	0.9868	0.9867	**1.2932**	**0.9862**	0.0069	0.0183	0.9749
PSGAN	**1.1740**	**0.9877**	**0.9880**	**1.2602**	**0.9869**	0.0067	**0.0116**	**0.9818**
FU-PSGAN	**1.2411**	**0.9869**	0.9865	**1.2907**	**0.9864**	0.0104	**0.0149**	0.9749
ST-PSGAN	1.2889	**0.9869**	**0.9868**	1.3267	0.9857	0.0138	0.0162	0.9702

TABLE V

PERFORMANCE COMPARISONS ON THE TEST SET OF GF-2. THE TOP-THREE PERFORMANCES ARE HIGHLIGHTED WITH **RED**, **GREEN**, AND **BLUE**

	SAM↓	CC↑	sCC↑	ERGAS↓	Q4↑	$D_\lambda \downarrow$	$D_S \downarrow$	QNR↑
SFIM [66]	1.5584	0.8721	0.9512	2.8705	0.8786	0.0123	0.0446	0.9437
LMVM [67]	2.0111	0.9073	0.9365	2.3138	0.9037	0.0022	0.0304	0.9675
LMM [67]	1.5527	0.8406	0.9450	3.0812	0.8387	0.0151	0.0508	0.9349
HPF [68]	1.5642	0.8776	0.9645	2.7818	0.8779	0.0118	0.0425	0.9462
HPFC [68]	1.7647	0.8852	0.9600	3.9200	0.8764	0.0840	0.0899	0.8337
Brovey [69]	1.3407	0.7990	0.9049	3.2624	0.8056	0.0454	0.1693	0.7930
HCS [70]	1.3407	0.8376	0.9392	3.2678	0.8296	0.0200	0.0615	0.9197
IHS [11]	1.8277	0.8109	0.9236	3.3495	0.8168	0.0530	0.1617	0.7938
GS [71]	2.2288	0.7898	0.8947	3.4247	0.7861	0.0819	0.1736	0.7587
BDSD [33]	1.8392	0.8791	0.9512	2.8705	0.8786	0.0066	0.0523	0.9415
PNN [9]	1.1899	0.9749	0.9821	1.2172	0.9946	0.0111	0.0494	0.9400
PanNet [10]	0.9370	0.9864	0.9889	0.8902	0.9971	0.0051	0.0128	0.9822
RED-cGAN [72]	**0.7442**	**0.9909**	**0.9931**	0.7223	0.9981	0.0005	**0.0088**	0.9908
PSGAN	0.7575	**0.9909**	0.9929	**0.7233**	**0.9980**	**0.0019**	**0.0060**	**0.9921**
FU-PSGAN	**0.7181**	**0.9915**	**0.9935**	**0.7013**	**0.9982**	**0.0020**	0.0089	**0.9892**
ST-PSGAN	**0.7300**	**0.9913**	**0.9933**	**0.7084**	**0.9981**	**0.0008**	**0.0070**	**0.9922**

TABLE VI

PERFORMANCE COMPARISONS ON THE TEST SET OF WV-2. THE TOP-THREE PERFORMANCES ARE HIGHLIGHTED WITH **RED**, **GREEN**, AND **BLUE**

	SAM↓	CC↑	sCC↑	ERGAS↓	Q4↑	$D_\lambda \downarrow$	$D_S \downarrow$	QNR↑
SFIM [66]	1.3411	0.9869	0.9892	3.5874	0.9873	**0.0016**	**0.0048**	**0.9936**
LMVM [67]	1.5580	0.9895	0.9874	3.2472	0.9897	0.0024	0.0053	0.9923
LMM [67]	1.5427	0.9890	0.9879	3.2039	0.9898	0.0062	0.0081	0.9857
HPF [68]	1.4367	0.9890	0.9890	3.2330	0.9889	**0.0017**	0.0049	0.9934
HPFC [68]	2.6736	0.9352	0.9670	7.8543	0.9186	0.0144	0.0342	0.9520
Brovey [69]	1.4023	0.9896	0.9890	3.1459	0.9891	0.0250	0.0171	0.9583
HCS [70]	1.4022	0.9895	0.9893	3.1017	0.9900	0.0118	0.0127	0.9757
IHS [11]	1.7003	0.9859	0.9895	3.5379	0.9890	0.0354	0.0245	0.9409
GS [71]	1.4448	0.9889	0.9878	3.2192	0.9879	0.0100	0.0166	0.9735
BDSD [33]	1.6422	0.9886	0.9924	3.3940	0.9905	0.0019	0.0052	0.9929
PNN [9]	1.4746	0.9955	0.9950	2.1630	0.9951	0.0086	0.0120	0.9795
PanNet [10]	0.9810	0.9966	0.9966	1.8530	**0.9964**	0.0044	0.0103	0.9854
RED-cGAN [72]	**0.8910**	**0.9973**	**0.9974**	**1.6639**	**0.9970**	**0.0013**	0.0049	**0.9938**
PSGAN	0.9127	**0.9973**	**0.9975**	**1.6452**	**0.9971**	0.0021	**0.0045**	0.9934
FU-PSGAN	**0.8855**	**0.9974**	**0.9974**	**1.6319**	**0.9971**	**0.0016**	**0.0038**	**0.9947**
ST-PSGAN	**0.9280**	**0.9972**	**0.9973**	1.6872	**0.9970**	0.0018	0.0085	0.9897

HPF [68], HPFC [68], Brovey [69], HCS [70], IHS [11], GS [71], BDSD [33], and three deep learning-based methods: PNN [9], PanNet [10], and RED-cGAN [72]. Tables IV–VI list the quantitative evaluations on the three data sets. Tables IV and V report the quality indexes of all comparison methods. It can be seen that deep models achieve surprisingly good performances and are superior to traditional methods in most cases. PanNet [10] is a successful method with very promising results. It obtains the best SAM on QB images and generalizes well to full-scale images, which is supported by its optimal nonreference metrics. As a pioneering deep model, PNN [9] proves the effectiveness of applying deep neural networks to pan-sharpening tasks. Although PNN has the lowest spectral quality on the QB data set, it works well on the GF-2 images with remarkable SAM surpassing all traditional methods. The proposed PSGAN obtains the best metrics on the QB images except for the SAM indicator. On the GF-2 data set, PSGAN and its variants show superior performance to all

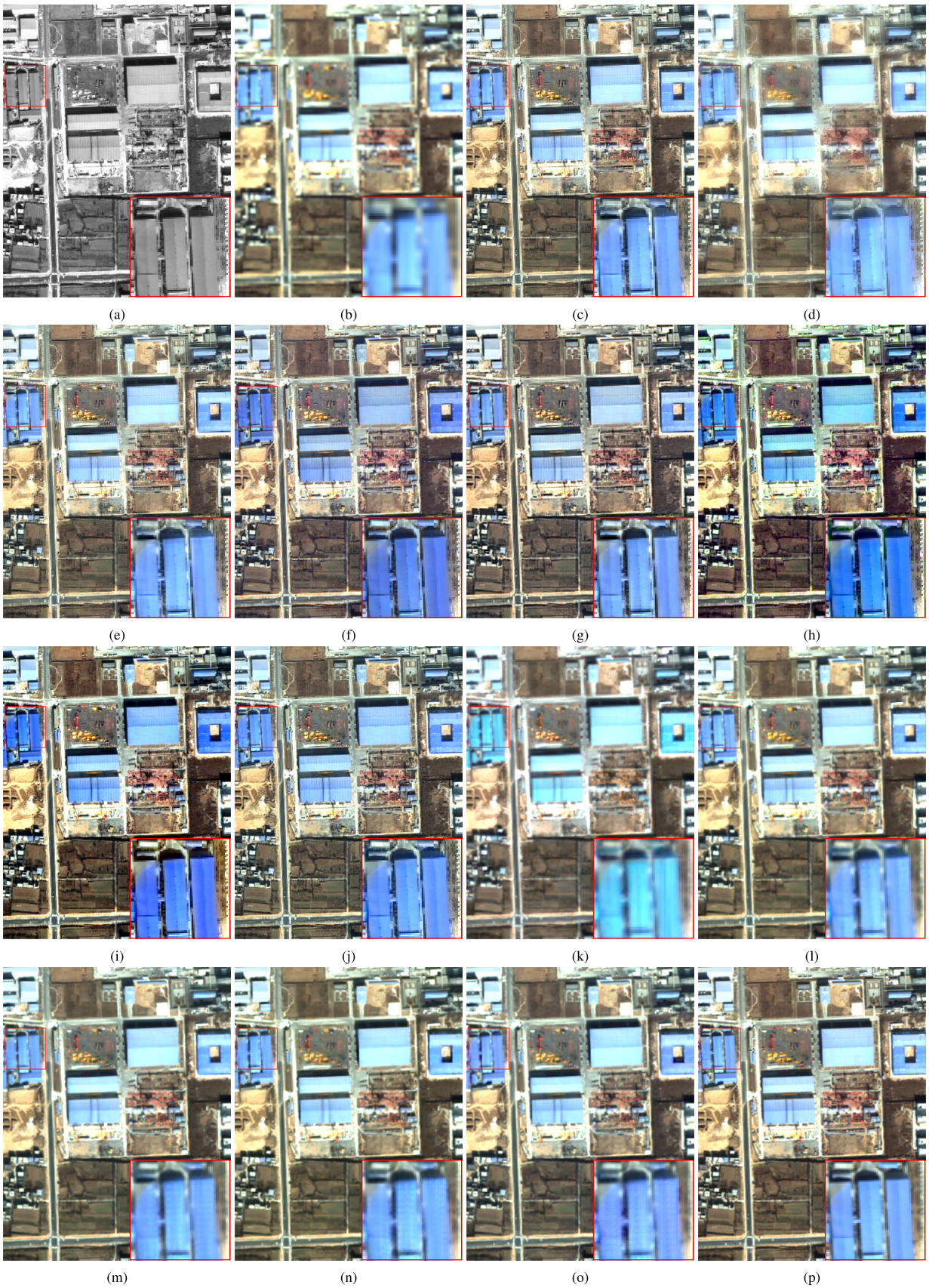

Fig. 5. Visual comparison on QB images. Images are displayed in RGB combination. All images have the same size of 465 × 360 pixels. (a) PAN. (b) LR MS. (c) SFIM [66]. (d) LMVM [67]. (e) HPF [68]. (f) Brovey [69]. (g) HCS [70]. (h) IHS [11]. (i) GS [71]. (j) BDSD [33]. (k) PNN [9]. (l) PanNet [10]. (m) PSGAN. (n) FU-PSGAN. (o) ST-PSGAN. (p) GT.

Fig. 6. Visual comparison on GF-2 images. Images are displayed in RGB combination. All images have the same size of 465 × 360 pixels. (a) PAN. (b) LR MS. (c) SFIM [66]. (d) LMVM [67]. (e) HPF [68]. (f) Brovey [69]. (g) HCS [70]. (h) IHS [11]. (i) GS [71]. (j) BDSD [33]. (k) PNN [9]. (l) PanNet [10]. (m) PSGAN. (n) FU-PSGAN. (o) ST-PSGAN. (p) GT.

TABLE VII

COMPUTATIONAL COSTS AND THE NUMBER OF PARAMETERS OF DIFFERENT MODELS ON THE TEST SETS. NOTE THAT THE PAN-SHARPENED IMAGES ARE WITH SIZES OF AROUND $3000 \times 2048 \times 4$, AND WE GIVE AVERAGE TIME ON THEM

Processor	Method	GFLOPS	Time (s)	♯Params
Intel Core i7-7700HQ CPU@2.80GHz	SFIM [66]	-	3.54	-
	LMVM [67]	-	22.60	-
	LMM [67]	-	3.57	-
	HPF [68]	-	3.66	-
	HPFC [68]	-	3.10	-
	Brovey [69]	-	0.41	-
	HCS [70]	-	4.13	-
	IHS [11]	-	0.39	-
	GS [71]	-	3.90	-
	BDSD [33]	-	40.36	-
NVIDIA GeForce RTX 2080Ti	PNN [9]	~ 84	0.43	~ 0.080 M
	PanNet [10]	~ 80	0.53	~ 0.077 M
	RED-cGAN [72]	~ 603	1.35	~ 1.90 M
	PSGAN	~ 402	1.13	~ 1.88 M
	PSGAN-f16	~ 97	0.62	~ 0.47 M
	PSGAN-k5$\times$5	~ 1011	1.52	~ 4.74 M
	FU-PSGAN	~ 392	1.07	~ 1.89 M
	ST-PSGAN	~ 351	0.98	~ 1.77 M

other methods. Especially, increasing the spatial resolution of MS images using CNN networks, i.e., FU-PSGAN, gains the best performance. Stacking the MS and PAN images together to perform pan-sharpening achieves the second-best place on the GF-2 data set; however, it falls behind the other two PSGAN models. Table VI presents the quantitative results of deep models on WV-2 images. As can be observed, our models still perform better than PNN [9] and PanNet [10]. Especially, FU-PSGAN achieves the best results on this data set with the highest SAM, CC, ERGAS, and Q_4 and slightly worse sCC than PSGAN.

I. Visual Comparisons

Figs. 5 and 6 show the sample results that are cropped from the test site of the Quickbird and GF-2 data sets, respectively. All images are displayed in true color. In Figs. 5 and 6, LMVM [67] and PNN [9] tend to blur images with very poor visual quality [see Figs. 5(d) and (k) and 6(d) and (k)]. SFIM [66], Brovey [69], HCS [70], IHS [11], GS [71], and BDSD [33] perform spatial information injection efficiently and produce results with clean high-frequency details almost identical to the PAN images. However, they suffer from severe spectral distortions, especially Brovey [see Fig. 5(f)], IHS [see Fig. 5(h)], and GS [see Fig. 5(i)] methods, the colors of which are darker than GT and MS images on the QB test set. Brovey [see Fig. 6(f)], HCS [see Fig. 6(g)], IHS [see Fig. 6(h)], GS [see Fig. 6(i)], and BDSD [see Fig. 6(j)] show noticeable color distortions on the GF-2 data set. The learning-based methods, i.e., PNN, PanNet, and ours, are optimized to generate images as close as to the GT images; thus, they have better results when comparing with the GT images. The proposed PSGANs produce results most similar to the GTs [see Figs. 5(m)–(o) and 6(m)–(o)]. One notable drawback of our methods and the other deep models is that they tend to produce smoother results than traditional ones, as can be seen from Figs. 5 and 6. This is mainly because of the pixelwise average problem [31] introduced by the loss function involving averaging operation, such as L_2. This phenomenon is frequently observed in image enhancement tasks. One possible solution to this is by using perceptual losses [77], which will be considered in our future work.

J. Experiments on Full-Resolution Images

We also evaluate our models on full-scale images without downsampling them. It should be noted that, under this setting, there will be no target images available for training. Considering that generalization ability across scales is the main concern in this experiment, we directly apply the optimized networks to the original PAN and MS images to produce the desired HR MS images. For quantitative evaluation, we calculate nonreference indexes as described in (12)–(14) for each pan-sharpened image and report results on the right sides of Tables IV–VI. As can be seen, the proposed PSGANs generalize well to the full-scale images. They obtain competitive performance on the three data sets. Especially, FU-PSGAN achieves the best results on WV-2 images; some typical samples are represented in Fig. 7 that clearly shows appealing results of FU-PSGAN.

K. Computational Time

We test the computational time of ours and the other comparison methods. All traditional methods are implemented using MATLAB and run on an Intel Core i7-7700HQ CPU, and deep models are implemented in PyTorch and tested on a single NVIDIA GeForce RTX 2080Ti GPU. The computational times are computed on the test sets of the three data sets. We give average time on the test sets for each method. Traditional methods are much faster than deep learning models. IHS [13] and Brovey [69] are the fastest ones. It takes less about 0.5 s for them to produce pan-sharpened images with sizes of about $3000 \times 2048 \times 4$. SFIM [66], LMM [67], HPF [68], HPFC [68], and GS [71] have almost the same time. They spend about $3 \sim 4$ s to pan-sharpen one image. HCS [70] takes a little longer; it takes more than 4 s to process one image. LMVM [67] and BDSD [33] are among the most time-consuming pan-sharpening. They spend 22 and 40 s, respectively, for generating one image. Beneficial from the advance of GPU architectures, deep learning-based

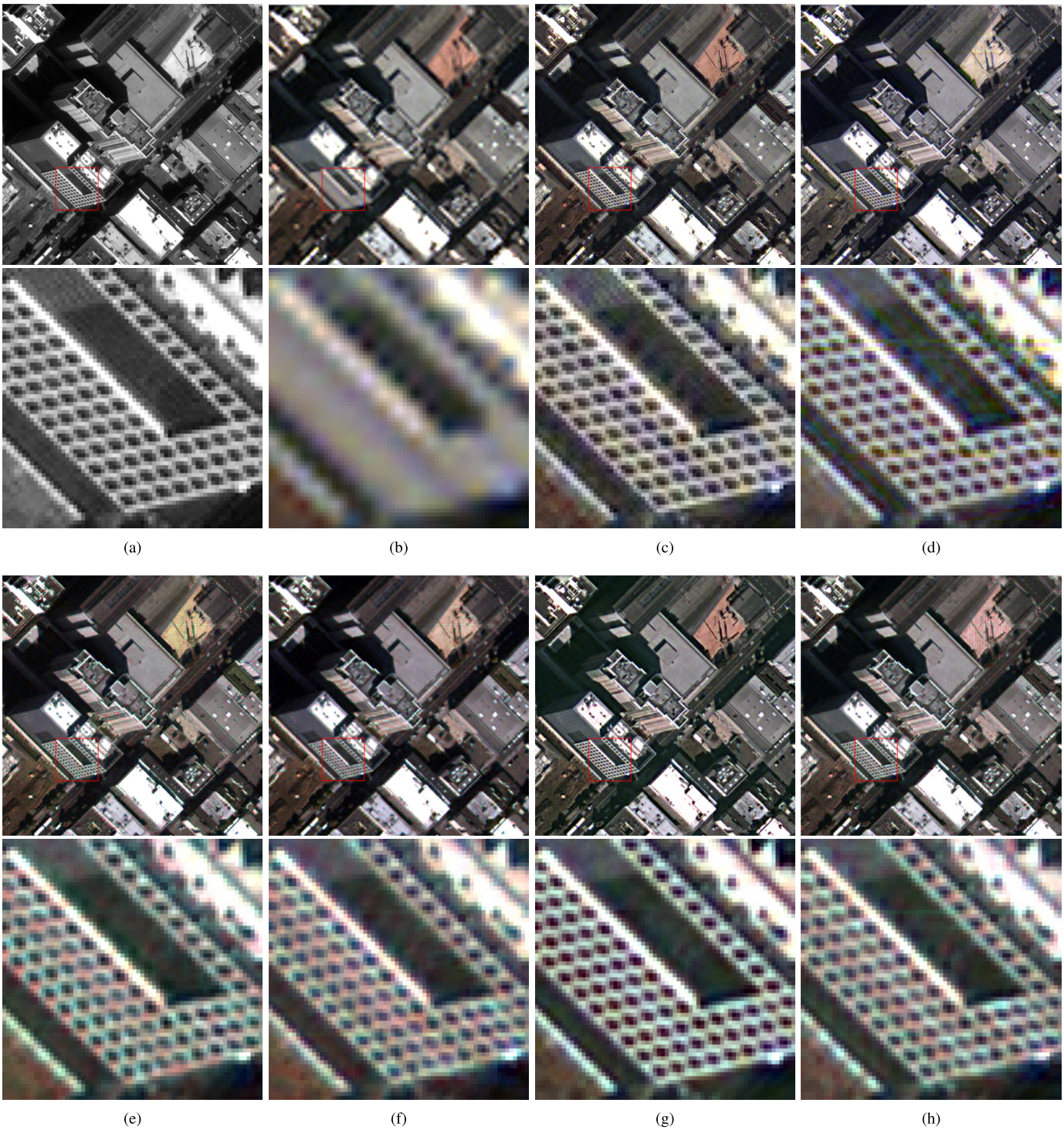

Fig. 7. Example results on WV-2 images (400×400 pixels). Displayed in RGB channels. (a) PAN. (b) LR MS. (c) SFIM [66]. (d) PNN [9]. (e) PanNet [10]. (f) PSGAN. (g) FU-PSGAN. (h) ST-PSGAN.

models are satisfactory. PNN [9] and PanNet [10] take about 0.43 and 0.53 s to process one image. It costs about 1.13, 1.07, and 0.98 s for PSGAN, FU-PSGAN, and ST-PSGAN to pan-sharpen one image. Our models are slower than PNN and PanNet because we have deeper architectures than them. FU-PSGAN performs a bit faster than PSGAN because it has deconvolution operations so that the input size is smaller. RED-cGAN [72] takes longer than PSGAN because it has more parameters.

V. Conclusion

In this article, we have proposed PSGANs for solving the task of image pan-sharpening and conducted extensive experiments on Quckbird, GaoFen-2, and Worldview-2 images.

The experiments demonstrate that the PSGANs are effective in generating high-quality pan-sharpened images with fine spatial details and high-fidelity spectral information under both low- and full-scale image settings and are superior to many popular pan-sharpening approaches. Furthermore, we evaluate several designs, including two-stream input, stacking input, BN layer, and attention mechanism, to find the optimal solution for the pan-sharpening task. We find that the two-stream architecture is normally better than the stacking strategy, and the BN layer and the SA module are not welcome in pan-sharpening. We suggest removing them from networks when designing pan-sharpening models.

In our future work, we will focus on unsupervised learning for pan-sharpening. Although remarkable results have been achieved by PSGANs, their generalization ability to full-scale images is still underdeveloped. We intend to solve this problem under an unsupervised learning framework and optimize the models using only original PAN and MS images without any preprocessing steps.

REFERENCES

[1] D. J. Mulla, "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps," *Biosyst. Eng.*, vol. 114, no. 4, pp. 358–371, Apr. 2013.

[2] A. Shalaby and R. Tateishi, "Remote sensing and GIS for mapping and monitoring land cover and land-use changes in the northwestern coastal zone of Egypt," *Appl. Geography*, vol. 27, no. 1, pp. 28–41, Jan. 2007.

[3] Q. Weng, "Thermal infrared remote sensing for urban climate and environmental studies: Methods, applications, and trends," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 4, pp. 335–344, Jul. 2009.

[4] Y. Zhang, "Understanding image fusion," *Photogram. Eng. Remote Sens*, vol. 70, no. 6, pp. 657–661, 2004.

[5] C. Thomas, T. Ranchin, L. Wald, and J. Chanussot, "Synthesis of multispectral images to high spatial resolution: A critical review of fusion methods based on remote sensing physics," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1301–1312, May 2008.

[6] Q. Xu, B. Li, Y. Zhang, and L. Ding, "High-fidelity component substitution pansharpening by the fitting of substitution data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7380–7392, Nov. 2014.

[7] G. Vivone *et al.*, "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2565–2586, May 2015.

[8] H. Ghassemian, "A review of remote sensing image fusion methods," *Inf. Fusion*, vol. 32, pp. 75–89, Nov. 2016.

[9] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sens.*, vol. 8, no. 7, p. 594, Jul. 2016.

[10] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "PanNet: A deep network architecture for pan-sharpening," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5449–5457.

[11] S. Rahmani, M. Strait, D. Merkurjev, M. Moeller, and T. Wittman, "An adaptive IHS pan-sharpening method," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 746–750, Oct. 2010.

[12] T.-M. Tu, S.-C. Su, H.-C. Shyu, and P. S. Huang, "A new look at IHS-like image fusion methods," *Inf. Fusion*, vol. 2, no. 3, pp. 177–186, Sep. 2001.

[13] T.-M. Tu, P. S. Huang, C.-L. Hung, and C.-P. Chang, "A fast intensity-hue-saturation fusion technique with spectral adjustment for IKONOS imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 309–312, Oct. 2004.

[14] S. C. S. Pat S. Chavez, Jr., "Comparison of three different methods to merge multiresolution and multispectral data: LANDSAT TM and SPOT panchromatic: ABSTRACT," *AAPG Bull.*, vol. 74, no. 3, pp. 295–303, 1991.

[15] H. R. Shahdoosti and H. Ghassemian, "Combining the spectral PCA and spatial PCA fusion methods by an optimal filter," *Inf. Fusion*, vol. 27, pp. 150–160, Jan. 2016.

[16] C. A. Laben and B. V. Brower, "Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening," U.S. Patent 6 011 875, Jan. 4, 2000.

[17] T. Ranchin and L. Wald, "Fusion of high spatial and spectral resolution images: The ARSIS concept and its implementation," *Photogramm. Eng. Remote Sens.*, vol. 66, no. 1, pp. 49–61, Jan. 2000.

[18] P. S. Pradhan, R. L. King, N. H. Younan, and D. W. Holcomb, "Estimation of the number of decomposition levels for a wavelet-based multiresolution multisensor image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3674–3686, Dec. 2006.

[19] J. Nunez, X. Otazu, O. Fors, A. Prades, V. Pala, and R. Arbiol, "Multiresolution-based image fusion with additive wavelet decomposition," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1204–1211, May 1999.

[20] F. Nencini, A. Garzelli, S. Baronti, and L. Alparone, "Remote sensing image fusion using the curvelet transform," *Inf. Fusion*, vol. 8, no. 2, pp. 143–156, Apr. 2007.

[21] C. Chen, Y. Li, W. Liu, and J. Huang, "Image fusion with local spectral consistency and dynamic gradient sparsity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2760–2765.

[22] X. He, L. Condat, J. M. Bioucas-Dias, J. Chanussot, and J. Xia, "A new pansharpening method based on spatial and spectral sparsity priors," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4160–4174, Sep. 2014.

[23] G. Vivone *et al.*, "Pansharpening based on semiblind deconvolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1997–2010, Apr. 2015.

[24] S. Li and B. Yang, "A new pan-sharpening method using a compressed sensing technique," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 738–746, Feb. 2011.

[25] X. X. Zhu and R. Bamler, "A sparse image fusion algorithm with application to pan-sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2827–2836, May 2013.

[26] X. X. Zhu, C. Grohnfeldt, and R. Bamler, "Exploiting joint sparsity for pansharpening: The J-SparseFI algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2664–2681, May 2016.

[27] Q. Liu, Y. Wang, and Z. Zhang, "Pan-sharpening based on geometric clustered neighbor embedding," *Opt. Eng.*, vol. 53, no. 9, Sep. 2014, Art. no. 093109.

[28] 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.

[29] Q. Zhang, Y. Wang, Q. Liu, X. Liu, and W. Wang, "CNN based suburban building detection using monocular high resolution Google Earth images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 661–664.

[30] S. Sreehari, S. V. Venkatakrishnan, K. L. Bouman, J. P. Simmons, L. F. Drummy, and C. A. Bouman, "Multi-resolution data fusion for super-resolution electron microscopy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1084–1092.

[31] 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. 4681–4690.

[32] Q. Yuan *et al.*, "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sens. Environ.*, vol. 241, May 2020, Art. no. 111716.

[33] A. Garzelli, F. Nencini, and L. Capobianco, "Optimal MMSE pan sharpening of very high resolution multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 1, pp. 228–236, Jan. 2008.

[34] X. Otazu, M. Gonzalez-Audicana, O. Fors, and J. Nunez, "Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 10, pp. 2376–2385, Oct. 2005.

[35] J. Zhong, B. Yang, G. Huang, F. Zhong, and Z. Chen, "Remote sensing image fusion with convolutional neural network," *Sens. Imag.*, vol. 17, no. 1, p. 10, Dec. 2016.

[36] Y. Rao, L. He, and J. Zhu, "A residual convolutional neural network for pan-shaprening," in *Proc. Int. Workshop Remote Sens. Intell. Process. (RSIP)*, May 2017, pp. 1–4.

[37] Y. Wei, Q. Yuan, H. Shen, and L. Zhang, "Boosting the accuracy of multispectral image pansharpening by learning a deep residual network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1795–1799, Oct. 2017.

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

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[40] Q. Yuan, Y. Wei, X. Meng, H. Shen, and L. Zhang, "A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 978–989, Mar. 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[43] W. Xie, Y. Cui, Y. Li, J. Lei, Q. Du, and J. Li, "HPGAN: Hyperspectral pansharpening using 3-D generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, early access, May 20, 2020, doi: 10.1109/TGRS.2020.2994238.

[44] X. Liu, Y. Wang, and Q. Liu, "Psgan: A generative adversarial network for remote sensing image pan-sharpening," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1–5.

[45] J. Xie *et al.*, "Pan-sharpening based on nonparametric Bayesian adaptive dictionary learning," in *Proc. ICIP*, 2013, pp. 2039–2042.

[46] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[47] X. Liu, Q. Liu, and Y. Wang, "Remote sensing image fusion based on two-stream fusion network," *Inf. Fusion*, vol. 55, pp. 1–15, Mar. 2020.

[48] Y. Qu, Y. Chen, J. Huang, and Y. Xie, "Enhanced Pix2pix dehazing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8160–8168.

[49] H. Yang, D. Huang, Y. Wang, and A. K. Jain, "Learning continuous face age progression: A pyramid of GANs," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 25, 2019, doi: 10.1109/TPAMI.2019.2930985.

[50] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 391–407.

[51] L. A. Maas, Y. A. Hannun, and Y. A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013, p. 3.

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

[53] L. Wald, "Quality of high resolution synthesised images: Is there a simple criterion?" in *Proc. Fusion Earth Data, Merging Point Meas., Raster Maps, Remotely Sensed Image*, 2000, pp. 99–103.

[54] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[56] R. H. Yuhas, A. F. H. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. Summaries 3rd Annu. JPL Airborne Geosci. Workshop*, Jun. 1992, pp. 147–149.

[57] J. Zhou, D. L. Civco, and J. A. Silander, "A wavelet transform method to merge landsat TM and SPOT panchromatic data," *Int. J. Remote Sens.*, vol. 19, no. 4, pp. 743–757, Jan. 1998.

[58] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.

[59] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, vol. 63, no. 6, pp. 691–699, 1997.

[60] L. Alparone, B. Aiazzi, S. Baronti, A. Garzelli, F. Nencini, and M. Selva, "Multispectral and panchromatic data fusion assessment without reference," *Photogramm. Eng. Remote Sens.*, vol. 74, no. 2, pp. 193–200, Feb. 2008.

[61] P. Goyal *et al.*, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*. [Online]. Available: http://arxiv.org/abs/1706.02677

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[63] G. Scarpa, S. Vitale, and D. Cozzolino, "Target-adaptive CNN-based pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5443–5457, Sep. 2018.

[64] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5345–5355, Nov. 2018.

[65] 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 (CVPRW)*, Jul. 2017, pp. 136–144.

[66] J. G. Liu, "Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details," *Int. J. Remote Sens.*, vol. 21, no. 18, pp. 3461–3472, Jan. 2000.

[67] S. D. Béthune, F. Müller, and J.-P. Donnay, "Fusion of multispectral and panchromatic images by local mean and variance matching filtering techniques," in *Proc. Fusion Earth Data*, Jan. 1998, pp. 28–30.

[68] U. G. Gangkofner, P. S. Pradhan, and D. W. Holcomb, "Optimizing the high-pass filter addition technique for image fusion," *Photogramm. Eng. Remote Sens.*, vol. 74, no. 9, pp. 1107–1118, Sep. 2008.

[69] A. R. Gillespie, A. B. Kahle, and R. E. Walker, "Color enhancement of highly correlated images. II. Channel ratio and 'chromaticity' transformation techniques," *Remote Sens. Environ.*, vol. 22, no. 3, pp. 343–365, Aug. 1987.

[70] C. Padwick, M. Deskevich, F. Pacifici, and S. Smallwood, "Worldview-2 pan-sharpening," in *Proc. ASPRS Annu. Conf.*, vol. 2630, San Diego, CA, USA, 2010, pp. 1–14.

[71] C. A. Laben and B. V. Brower, "Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening," U.S. Patent 6 011 875, Jan. 4, 2000.

[72] Z. Shao, Z. Lu, M. Ran, L. Fang, J. Zhou, and Y. Zhang, "Residual encoder–decoder conditional generative adversarial network for pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 9, pp. 1573–1577, Sep. 2020.

[73] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[74] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[75] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7354–7363.

[76] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[77] 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.

Qingjie Liu (Member, IEEE) received the B.S. degree in computer science from Hunan University, Changsha, China, and the Ph.D. degree in computer science from Beihang University, Beijing, China.

He is an Associate Professor with the School of Computer Science and Engineering, Beihang University. He is also a Distinguished Research Fellow with the Hangzhou Institute of Innovation, Beihang University, Hangzhou. His research interests include image fusion, object detection, image segmentation, and change detection.

Huanyu Zhou received the B.S. degree in computer science from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2020, where he is pursuing the M.S. degree with the Laboratory of Intelligent Recognition and Image Processing, School of Computer Science and Engineering, Beihang University.

His research interests include computer vision and pattern recognition.

Qizhi Xu (Member, IEEE) received the B.S. degree from Jiangxi Normal University, Nanchang, China, in 2005, and the Ph.D. degree from Beihang University, Beijing, China, in 2012.

He was a Post-Doctoral Fellow with the University of New Brunswick, Fredericton, NB, Canada. He is an Associate Professor of the Beijing Institute of Technology, School of Mechatronical Engineering, Beijing, China. His research interests include image fusion, image understanding, and big data analysis of remote sensing.

Dr. Xu was a recipient of the Technological Invention Award First Prize from the Chinese Institute of Electronics for his image fusion research in 2017.

Xiangyu Liu received the B.S. and M.S. degrees in computer science from the School of Computer Science and Engineering, Beihang University, Beijing China, in 2015 and 2018, respectively.

His research interests include computer vision and deep learning.

Yunhong Wang (Fellow, IEEE) received the B.S. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1989, and the M.S. and Ph.D. degrees in electronic engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1995 and 1998, respectively.

She was with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 1998 to 2004. Since 2004, she has been a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, where she is also the Director of the Laboratory of Intelligent Recognition and Image Processing. Her research interests include biometrics, pattern recognition, computer vision, data fusion, and image processing.

Dr. Wang is a Fellow of IAPR and CCF.