Spatial Data Augmentation: Improving the Generalization of Neural Networks for Pansharpening

Lihui Chen, *Member, IEEE*, Gemine Vivone, *Senior Member, IEEE*, Zihao Nie, Jocelyn Chanussot, *Fellow, IEEE*, and Xiaomin Yang, *Member, IEEE*

***Abstract*— Deep learning (DL) methods have achieved impressive performance for pansharpening in recent years. However, because of poor generalization, most DL methods achieve unsatisfactory performance for data acquired by sensors not considered during the training phase and decreased performance for samples at full resolution (FR). To solve this issue, we propose a data augmentation framework for pansharpening neural networks (PNNs). Specifically, we introduce first a random spatial degradation based on anisotropic Gaussian-shaped modulation transfer functions (MTFs) to increase the generalization with respect to different spatial models and sensors. Then, considering that various sensors have different ground sampling distances (GSDs), we randomly rescale the GSD of the training samples to improve the generalization with respect to spatial resolution. Thanks to this module, the generalization to tests from different sensors and samples at FR can easily be achieved. Experimental results demonstrate the effectiveness of the proposed approach with better performance when data for training are decoupled with the ones for testing and comparable performance when training and testing are coupled (i.e., data acquired by the same sensor are considered in the two phases). Besides, performance at FR for PNNs is improved by the proposed approach. The proposed approach has been integrated into existing PNNs showing satisfactory performance for widely used sensors, including, GaoFen-1 (GF1), QuickBird (QB), WorldView-2 (WV2), WorldView-3 (WV3), IKONOS (IK), Spot-7, GeoEye, and PHR1A.**

***Index Terms*— Convolutional neural networks (CNNs), data augmentation, data fusion, deep learning (DL), multispectral imaging, pansharpening, remote sensing.**

Manuscript received 19 January 2023; revised 20 March 2023; accepted 21 March 2023. Date of publication 27 March 2023; date of current version 11 April 2023. This work was supported in part by the China Scholarship Council (CSC) under Grant 202006240191, in part by the Science Foundation of Sichuan Science and Technology Department under Grant 2021YFH0119, and in part by Sichuan University under Grant 2020SCUNG205. *(Corresponding author: Xiaomin Yang.)*

Lihui Chen is with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: lihui.chen@cqu.edu.cn).

Gemine Vivone is with the National Research Council, Institute of Methodologies for Environmental Analysis, CNR-IMAA, 85050 Tito, Italy, and also with the National Biodiversity Future Center (NBFC), 90133 Palermo, Italy (e-mail: gemine.vivone@imaa.cnr.it).

Zihao Nie and Xiaomin Yang are with the College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China (e-mail: niezihao@stu.scu.edu.cn; arielyang@scu.edu.cn).

Jocelyn Chanussot is with the Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Laboratory, 38000 Grenoble, France (e-mail: jocelyn.chanussot@grenoble-inp.fr).

Digital Object Identifier 10.1109/TGRS.2023.3262262

I. Introduction

PANSHARPENING aims at fusing the high-frequency details from a high-resolution panchromatic (PAN) image into a low-resolution multispectral (LRMS) image with finer spectral information, thereby obtaining a high-resolution multispectral (HRMS) image.

Recently, deep learning (DL) achieved good performance for pansharpening. However, the generalization is still an open problem for pansharpening neural networks (PNNs), which limits the applicability of PNNs in real contexts. Specifically, the poor generalization leads to the following issues. Almost all DL methods have to train separate models for different sensors, which is time- and resource-consuming. This is a straightforward drawback compared with traditional methods that can process data from any sensor by a single model. Besides, we may lack training samples from a specific sensor, in which case a DL model trained by data from other sensors with advanced generalization can be applied to the specific sensor. Moreover, the generalization of DL methods for full resolution (FR) samples is still a challenge since most of DL methods are trained at reduced resolution (RR), thus showing a performance drop at FR. Therefore, it is crucial the study of the generalization of PNNs.

There are two problems when fusing MS and PAN images acquired by various sensors using a single DL model. The first one is the band number of MS images. MS images from different sensors may have different numbers of bands, such as four bands for QuickBird (QB), eight bands for WorldView-2/-3 (WV2/WV3), six bands with 20-m ground sampling distance (GSD) for Sentinel-2 MS images [1], three bands with 60-m GSD for Sentinel-2 MS images, and nine bands with 30-m GSD for ALI. For most DL methods, the limitation of processing MS images with a constant number of bands precludes the sharpening of MS images with a different number of bands. Although a model, named ArbRPN, has been proposed in [2] to process MS images with arbitrary numbers of bands, the network still cannot obtain a satisfactory performance for samples acquired by sensors not presented in the training set, thus showing a limited generalization. Unfortunately, there is almost no literature to study the generalization for PNNs. Although Scarpa et al. [3] proposed a target-adaptive fine-tuning to process samples out of the training set, fine-tuning

cannot improve the generalization of a model, but it trains an image-specific model. Differently, we aim at improving the generalization of PNNs by means of data augmentation, and then combining the augmentation strategy with existing networks.

To improve the performance at FR, some methods [4], [5], [6], [7], [8], [9] removed the resolution gap by directly training models at FR, whereas it is nontrivial to train a model at FR in a stable way. Different from these methods, we improve the performance at FR by increasing the generalization of PNNs at FR. We train the PNNs at RR by data augmentation based on random GSD, which is stable and performs well at FR.

It has been shown in many high-level visual tasks that data augmentation, such as Cutout [10], CutMix [11], MixUp [12], can improve the generalization of DL models. However, data augmentation for high-level visual tasks cannot directly be adopted to low-level ones [13]. Data augmentation for generalization seems to be neglected in the pansharpening community [13].

To fill this gap, a data augmentation framework is proposed in this article to improve the generalization for PNNs. We augment data in two ways. First, we consider that various sensors have different modulation transfer functions (MTFs). We utilize random MTFs to degrade MS images and generate the RR samples. This augmentation can increase the diversity of training samples, thus covering more spatial models for sensors. The Gaussian function is adopted to simulate random MTFs in our augmentation framework since it has a shape that is close to the real MTFs of sensors [14]. Besides, Aiazzi et al. [14] showed that a typical MTF for MS bands is anisotropic, thus driving us toward the simulation of anisotropic Gaussian-shaped MTFs. Second, we augment training samples by randomly rescaling the GSD of MS and PAN images, which can increase the GSD diversity of training samples. Various sensors usually have distinct GSDs. Even the same sensor has distinct GSDs for different bands, such as the aforementioned Sentinel-2. Moreover, the invariance among scales property for training DL approaches is not always valid. Thus, the model trained at RR could have lower performance at FR. Therefore, we think the GSD is also an important factor. Through augmenting the GSD diversity of training samples, a network can get improved performance even when fusing samples at different resolutions with respect to the observed ones during the training phase, as shown in Fig. 1 and in Section IV-B.

In summary, the contributions of this article are as follows.

1) We study the generalization of PNNs from the data augmentation point of view. A simple yet effective data augmentation framework is proposed to improve the generalization of existing PNNs.
2) The generation of anisotropic Gaussian-shaped MTFs and the rescaling of GSDs are proposed to improve the generalization of PNNs with respect to spatial model and resolution, respectively, thus also improving performance at FR.
3) The proposed approach has been integrated into existing PNNs and works well for widely used sensors using a single model. To the best of our knowledge, the combination with existing neural networks, such as, ArbRPN, represents the first DL-based solution to address the fusion task for many different sensors without any subsequent fine-tuning, architecture changing, or parameter tuning.

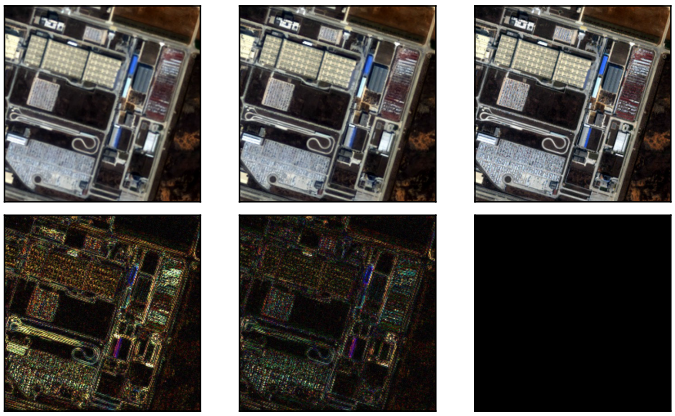


Fig. 1. Comparison between the model trained with and without random GSD rescaling. (Top row) RGB visualization. (Bottom row) Absolute error between the fused image and the GT. The LRMS image in the test phase has a GSD of 12.8 m, while samples for training have GSDs of 32, 9.76, and 7.36 m.

The rest of this article is organized as follows. Section II reviews some DL methods for pansharpening. The proposed augmentation framework is presented in Section III. Experimental results are conducted in Section IV. Finally, Section V draws the conclusion of this article.

II. Related Works

Masi et al. [15], inspired by the convolutional neural network (CNN) for image super-resolution [16], proposed the first CNN for pansharpening, named PNN. Its superior performance encouraged many DL methods. A target-adaptive PNN to adapt the model to the target image during the testing phase has been proposed in [3]. Wei et al. [17] introduced deep residual learning to learn residuals for input images, then generating fused results by a mapping of spectral dimensional reduction. A multiscale deep CNN has been proposed in [18] to exploit multiscale information for pansharpening. PanNet [19] fused the MS and PAN images in the high-pass filtering domain to pursue spectral preservation and spatial enhancement of LRMS images. Liu et al. [20] proposed a two-stream fusion network, in which features of MS and PAN images have been separately extracted and then fused into the feature domain. To better enhance details, Lai et al. [21] devised two branches for pansharpening, one of which using the gradient information to guide the fusion of MS and PAN images in the other branch. Assuming that MS bands are independent of one another, Liu et al. [22] proposed a band-independent encoder–decoder network (BIEDN) to process MS images with any number of bands. However, this assumption hardly holds. Thus, Chen et al. [2] proposed a bidirectional recurrent pansharpening network, named ArbRPN, for MS images with arbitrary numbers of bands. Formulating pansharpening as colorization of PAN images, Ozcelik et al. [23] proposed a generative adversarial network (GAN) to learn the relationship between the LRMS image and an intensity image synthesized by the

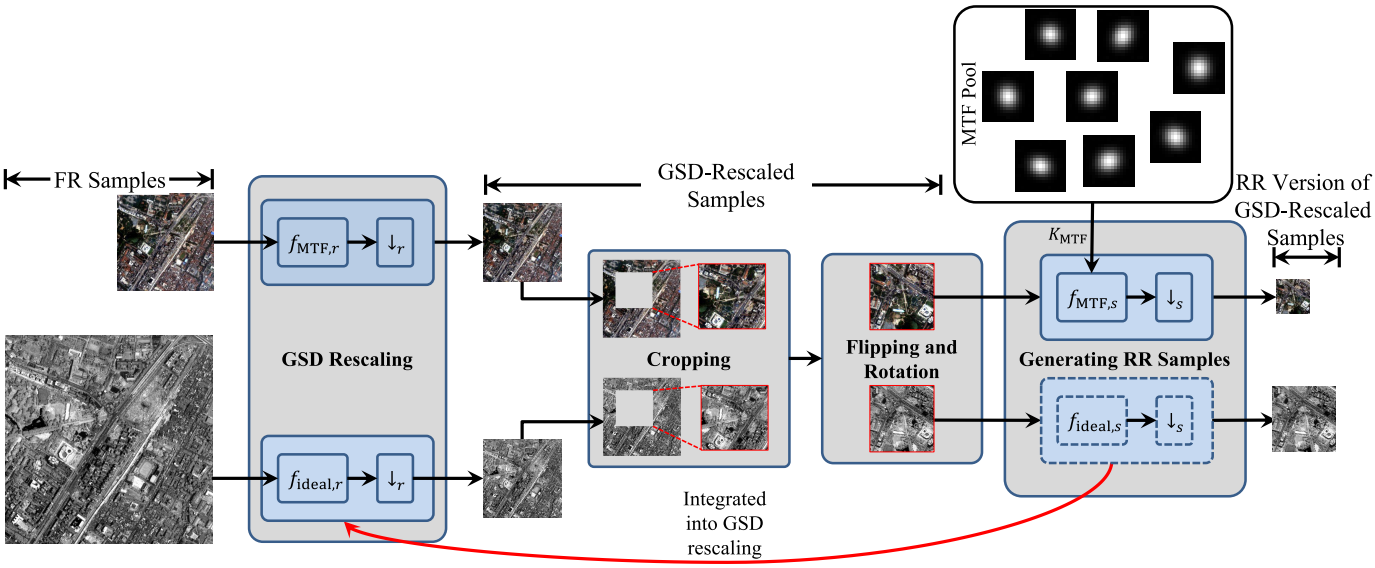


Fig. 2. Proposed data augmentation framework for pansharpening.

LRMS image, then applying it to PAN images to achieve the HRMS images. Finally, Lei et al. [24] proposed a nonlocal attention residual network to leverage on the self-similarity in images for pansharpening.

Recently, some unsupervised neural networks have also been presented. Unlike the above-mentioned supervised methods, these approaches directly train a model at FR, thus relaxing the assumption of invariance among scales. Luo et al. [4] proposed an unsupervised loss function based on the relationship between the HRMS image and the LR observations, i.e., PAN and LRMS data. Besides, an adjustment component based on the quality with no reference (QNR) index [25] has been introduced in the loss function for spectral and spatial preservation. Qu et al. [5] utilized an encoder built by stacked attention modules to extract attention representation for MS images, then injecting space-varying details by space-varying injection coefficients into the attention representation. Finally, an image is reconstructed by the decoder with the detail-injected attention representation. Ma et al. [6] introduced an unsupervised GAN for pansharpening. Two discriminators have been adopted into the network to enforce spectral and spatial preservation, respectively. Following the same research line, pansharpening generative multi-adversarial network (PGMAN) has been proposed by Zhou et al. [7] consisting of an unsupervised loss based on the QNR. In [8], a cycle-consistent loss has been introduced for pansharpening. Finally, Ciotola et al. [9] proposed a training framework at FR to avoid information loss in training data and to guarantee the performance at FR. Unlike the above methods, which adopt unsupervised training at FR, our approach can improve the generalization of neural networks at FR by supervised learning considering lower resolutions, thus generalizing with respect to the resolution and obtaining a training that is more stable than that of provided by unsupervised learning using loss functions including (usually inaccurate) evaluations at FR. Besides, PNNs trained by our data augmentation approach have a superior generalization to samples from different sensors.

III. Methodology

In this section, we will overview first the proposed framework. Then, the implementations of random anisotropic Gaussian-shaped MTFs and random GSD rescaling are introduced.

A. Data Augmentation Framework: An Overview

Fig. 2 shows the proposed augmentation framework, which consists of four steps: 1) random GSD rescaling; 2) random cropping; 3) random flipping and rotation; and 4) generation of RR samples.

First, we randomly rescale the GSD of the FR images to improve the generalization of networks in fusing images with different resolutions. A variable subject to an uniform distribution with interval of $[0, 1)$ is used to decide if MS and PAN images should be rescaled or not. When the random variable is smaller than a threshold, t (empirically set to 0.2 in this article), MS and PAN images are rescaled by

$$\mathbf{I}'_{\mathrm{MS},i} = \downarrow_r \left(f_{\mathrm{MTF},r}\left(\mathbf{I}_{\mathrm{MS},i}\right)\right) \tag{1}$$

$$\mathbf{P}' = \downarrow_r \left(f_{\mathrm{ideal},r}\left(\mathbf{P}\right)\right) \tag{2}$$

where $f_{\mathrm{MTF},r}(\cdot)$, and $f_{\mathrm{ideal},r}(\cdot)$ denote the MTF-based and ideal filtering with a normalized cut-off frequency dependent on r, respectively, $\downarrow_r (\cdot)$ is the decimation operation with decimated rate of $(1/r)$, $\mathbf{I}_{\mathrm{MS},i}$, and $\mathbf{I}'_{\mathrm{MS},i}$ are the ith band of the original MS and GSD-rescaled MS images, respectively, $\mathbf{P}$ and $\mathbf{P}'$ are the original and GSD-rescaled PAN images, respectively, $r \sim U(0.5, 1)$ is the random scale ratio used to change the GSD of MS and PAN images, and $U(a, b)$ refers to a uniform distribution with interval $[a, b)$.

Then, similar to general augmentation methods as in [2] and [21], a patch of predefined size, $p \times p$, is randomly cropped from the rescaled MS and PAN images. The left-upper corner of the crop position is controlled by two random variables $U(1, H - p)$ and $U(1, W - p)$, respectively, where H and W are the height and width of the corresponding image.

Afterward, the cropped patches are flipped vertically or horizontally, if two random variables sampled according to $U(0, 1)$ are greater than a threshold. The rotation of 90° is also imposed to the MS and PAN patches if another random variable sampled according to $U(0, 1)$ is greater than the same threshold.

Finally, the RR MS images are generated by random anisotropic Gaussian-shaped MTFs, while the RR PAN samples are achieved by ideal filtering like in [14], [26], and [27]. After this last step, we have the RR versions of the GSD-rescaled MS and PAN images. These images are taken as input for the training step, while the GSD-rescaled MS image before MTF-based degradation is taken as ground-truth (GT) for supervised learning.

Since misalignment can occur in the stage of GSD rescaling and generation of RR samples (see Section III-C), we moved the ideal downsampling for the generation of RR versions of GSD-rescaled PAN images to the GSD-rescaling stage combining them together, as shown by the red skip arrow in Fig. 2. Thus, the GSD-rescaling operation in (2) becomes a combined operation of GSD-rescaling and generation of RR samples, i.e.,

$$\widetilde{\mathbf{P}} = \downarrow_{\frac{r}{s}} \left(f_{\text{ideal}, \frac{r}{s}}(\mathbf{P}) \right) \tag{3}$$

where s is the scale ratio between MS and PAN and $\widetilde{\mathbf{P}}$ denotes the RR version of the GSD-rescaled PAN image.

B. Degradation With Random Anisotropic MTFs

MTF-based degradation has been used by widespread DL methods to generate RR samples for training. Generally, isotropic Gaussian-shaped MTF is used to degrade the original samples for two reasons: 1) its shape is close to the true MTF of an MS band and 2) given the scale ratio, it is easy to determine isotropic Gaussian-shaped MTFs by a sole parameter, i.e., the standard deviation, which can be derived from the gain at Nyquist frequency. However, Aiazzi et al. [14] demonstrated that the true MTF is usually anisotropic. Hence, the augmentation by random anisotropic MTFs can simulate a more practical degradation and further improve the diversity of training samples. Thus, we use anisotropic Gaussian functions to simulate the MTFs, although the augmentation by random isotropic Gaussian-shaped MTFs already obtain a satisfactory generalization (see Section IV-B). Specifically, an anisotropic Gaussian-shaped MTF can be determined by two standard deviations and a rotation degree. Therefore, we can use three independent random variables, σ_1, σ_2, and θ, to generate a random anisotropic Gaussian-shaped MTF. θ sampling from $U(0, \pi)$ is used to determine the rotation degree of the anisotropic Gaussian kernel. σ_1 and σ_2 are two standard deviations for the anisotropic Gaussian kernel.

For isotropic Gaussian-shaped MTFs, we can formulate the gain at Nyquist frequency considering a Gaussian function as following:

$$G = \exp\left(-\frac{\left(\frac{(N-1) f_{\text{cut}}}{2} \right)^2}{2\sigma^2} \right) \tag{4}$$

where G is the gain at Nyquist frequency, $f_{\text{cut}} = (1/s)$ is the normalized Nyquist frequency, N and σ are the support and the standard deviation of the Gaussian function, respectively. Therefore, given the scale ratio s and the support N, as long as we have the gain G of an imaging system, the standard deviations can be derived from the gain when its MTF is approximated by a Gaussian function. Thus, the standard deviations can be formulated as a function of the gain. Specifically, the relationship between the standard deviation and the gain at Nyquist frequency is as follows:

$$\sigma = \sqrt{\frac{\left(\frac{(N-1) f_{\text{cut}}}{2} \right)^2}{-2\ln(G)}}. \tag{5}$$

Like [26], and [28], N is set to 41. ln indicates the natural logarithm. In this article, we focus on pansharpening with a scale ratio s equal to 4. Therefore, for anisotropic Gaussian-shaped MTFs, we can achieve two random standard deviations, i.e., σ_1 and σ_2, by sampling two Gs from two independent uniform distributions, $U[0.2, 0.4)$. The reason of choosing this interval for G is that the gains at Nyquist frequency for well-known MS sensors are located in this interval. Given the standard deviations and the rotation degree, the Gaussian-shaped anisotropic MTF can be obtained by

$$K_{\text{MTF}}(x, y) = \exp\left(-\frac{1}{2} \mathbf{v} \Sigma \mathbf{v}^T \right) \tag{6}$$

where $\mathbf{v} = [x, y]$, $\cdot^T$ is the transpose operator, and Σ is a covariance matrix defined as

$$\Sigma = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{7}$$

Finally, a finite impulse response (FIR) filter is designed through the Kaiser window approach according to the randomly generated MTF. Applying it to an MS band and then decimating the outcome with a factor s, we obtain the RR MS band by the degradation of a random anisotropic Gaussian-shaped MTF.

C. Random Rescaling of GSD

The simulation of images captured at different resolutions is done according to Wald's protocol in the same way as the classical generation of products for the assessment at RR. In particular, an ideal filter is used for PAN images and MTF-based degradation is considered for MS images. The MTF-based degradation for MS data in this stage is similar to what we detailed in Section III-B. However, differently from Section III-B, isotropic Gaussian-shaped MTFs are used fixing the gains on a sensor basis as suggested in [27]. Only the scale ratio is sampled according to the uniform random variable $U(0.5, 1)$.

As mentioned in Section III-A, there may exist spatial misalignments between RR MS and PAN images, if we rescale first the GSD and then generating the RR samples. Specifically,

TABLE I
DETAILS OF THE DATASETS USED IN THE EXPERIMENTS. THE SIZE IS REFERRED TO MS IMAGES AT FR. #BANDS IS THE NUMBER OF BANDS FOR MS IMAGES

Datasets	For training, validation, test sets			Only for test sets				
	GF1	QB	WV2	IK	WV3	Spot-7	GE	PHR1A
#Samples	410	500	500	40	32	2	2	3
Size	512×512	512×512	512×512	512×512	512×512	512×592 480×672	512×512	1152×1856 2224×1552 1840×1296
#Bands	4	4	8	4	8	4	4	4
GSD (m)	8	2.44	1.84	3.2	1.84	6	1.84	2

given an MS image with height, h, and width, w, the size of the GSD-rescaled MS image is $\lceil hr \rceil$, $\lceil wr \rceil$, where r is the random scale ratio sampled from $U(0.5, 1)$. $\lceil \cdot \rceil$ denotes the ceil operation.[1] Similarly, given a PAN image with height, H, and width, W, the size of the GSD-rescaled PAN image is $\lceil Hr \rceil$, $\lceil Wr \rceil$. $H = sh$ and $W = sw$, where s is the scale ratio between MS and PAN images. If one requires that GSD-rescaled MS and PAN images are still coregistered, their size should satisfy the conditions

$$\lceil Hr \rceil = s \lceil hr \rceil, \tag{8}$$

$$\lceil Wr \rceil = s \lceil wr \rceil. \tag{9}$$

Moreover, the following conditions should be also valid to ensure that the RR versions of GSD-rescaled MS and PAN images are coregistered:

$$\left\lceil \frac{\lceil Hr \rceil}{s} \right\rceil = s \left\lceil \frac{\lceil hr \rceil}{s} \right\rceil \tag{10}$$

$$\left\lceil \frac{\lceil Wr \rceil}{s} \right\rceil = s \left\lceil \frac{\lceil wr \rceil}{s} \right\rceil. \tag{11}$$

However, these conditions are not always valid because r is a random ratio sampled according to an uniform distribution [i.e., $U(0.5, 1)$]. For example, when $H = W = 64$, $h = w = 16$, and $s = 4$, if r is 0.64, then the sizes of the GSD-rescaled PAN and MS images are 41×41 and 11×11, thus generating misalignments. The reason for setting the interval to $[0.5, 1)$ is that we empirically found that this interval achieves the best results among the ones in the set $\{[0.5, 1),\ [0.25, 1),\ [0.5, 1.5),\ [1, 2),\ [1, 4)\}$.

To avoid the misalignment between the RR versions of GSD-rescaled MS and PAN images, we integrate the ideal downsampling for the generation of RR PAN images into the GSD rescaling operation. That is, we generate the RR version of the GSD-rescaled PAN image by ideal downsampling with a scale ratio of (r/s), as shown in (3). Indeed, the image rescaled by (r/s) is very close to the image obtained by sequentially rescaling with factors r and $(1/s)$ when bicubic filtering[2] is applied to the image. This is due to the fact that the above-mentioned filter is an almost ideal filter.

Through (3), we have an RR version of the GSD-rescaled PAN image with size of $(\lceil (Hr/s) \rceil, \lceil (Wr/s) \rceil)$. Its size is the same as that of the GSD-rescaled MS image, i.e., $(\lceil hr \rceil, \lceil wr \rceil)$, since $H = sh$ and $W = sw$. Then, to enforce coregistration among products to be fused, we only need to make sure that $(\lceil hr \rceil / s)$ and $(\lceil wr \rceil / s)$ are integers. Random cropping in Fig. 2 can work in this direction, thus cropping patches with a size meeting the above conditions.

[1]Some other implementations can use round or floor operations.

[2]As proposed in previous works [14], [26], [27] for generating RR PAN images.

IV. EXPERIMENTAL ANALYSIS

In this section, we will introduce first the datasets and training details. Afterward, we will demonstrate the effectiveness of the key components of the proposed data augmentation framework and we will show the improvements of the framework for many well-known PNNs. Finally, we will compare ArbRPN trained by our data augmentation (ArbRPN+ for short) with the state-of-the-art (SOTA) methods from both quantitative and qualitative perspectives. Two test cases are considered: 1) *known sensors*, i.e., testing images acquired by a sensor considered during the training and 2) *unknown sensors*, i.e., testing images acquired by a sensor not considered during the training.

A. Datasets and Training Details

1) Datasets: We adopted five datasets from [29] and three datasets from [27] involving the following sensors: QB, WV2, GaoFen-1 (GF1), WV3, IKONOS (IK), PHR1A, Spot-7, and GeoEye-1 (GE1). WV3, IK, PHR1A, Spot-7, and GE1 are only used for test sets to show the sensor-based generalization ability, and the other three datasets are used for training, validation, and test sets. MS images for WV2 and WV3 have eight bands, while MS images for the other sensors have four bands. The numbers of MS-PAN pairs for each dataset are shown in Table I. We randomly split these samples into training and test sets by a ratio of 8:2. Then, 20 images from each training set are randomly selected as validation set. To obtain RR samples for validation and test sets, we adopt Wald's protocol using the protocol described in [27]. The original MS and PAN images are adopted for tests at FR. The training samples are obtained by applying the proposed data augmentation framework to images belonging to a (mixing) dataset consisting of images acquired by three sensors: GF1, QB, and WV2.

2) Training: The ℓ_1 loss is used for training all the networks. The models are trained by 10^5 iterations, while the models in Table II and Fig. 3 are trained by 2×10^4 for time saving. The Adam [30] algorithm is used to optimize these networks with Pytorch's default settings, except for the initial learning rate (set to 0.0002). The learning rate reduces to half

TABLE II
AVERAGE RESULTS ON *Known* AND *Unknown Sensors* FOR MODELS TRAINED BY DIFFERENT AUGMENTATION STRATEGIES

Methods	Average on *known sensors*							Average on *unknown sensors*						
	SAM	ERGAS	Q2n	sCC	D_λ	D_s	HQNR	SAM	ERGAS	Q2n	sCC	D_λ	D_s	HQNR
Mix	1.7984	1.6816	0.8326	0.9692	0.0609	0.0800	0.8705	3.8324	3.9806	0.7834	0.8866	0.0790	0.0972	0.8362
Mix+Norm	1.7686	1.6501	0.8344	0.9711	0.0441	**0.0673**	0.8943	3.4511	4.0542	0.7938	0.8785	0.0450	0.0943	0.8657
Mix+Norm+RMTF	1.6885	1.5351	0.8462	0.9755	0.0301	0.0734	0.9009	3.3077	3.6087	0.8190	0.8904	0.0239	0.0996	0.8791
Mix+Norm+RMTF+AN	**1.6787**	**1.5123**	**0.8536**	**0.9774**	0.0282	0.0890	0.8876	3.2057	3.5614	0.8215	0.9187	0.0249	0.1049	0.8728
Mix+Norm+RMTF+AN+SC	1.7302	1.5830	0.8434	0.9753	**0.0199**	0.0727	**0.9099**	**3.1545**	**3.0895**	**0.8413**	**0.9337**	**0.0206**	**0.08980**	**0.8917**

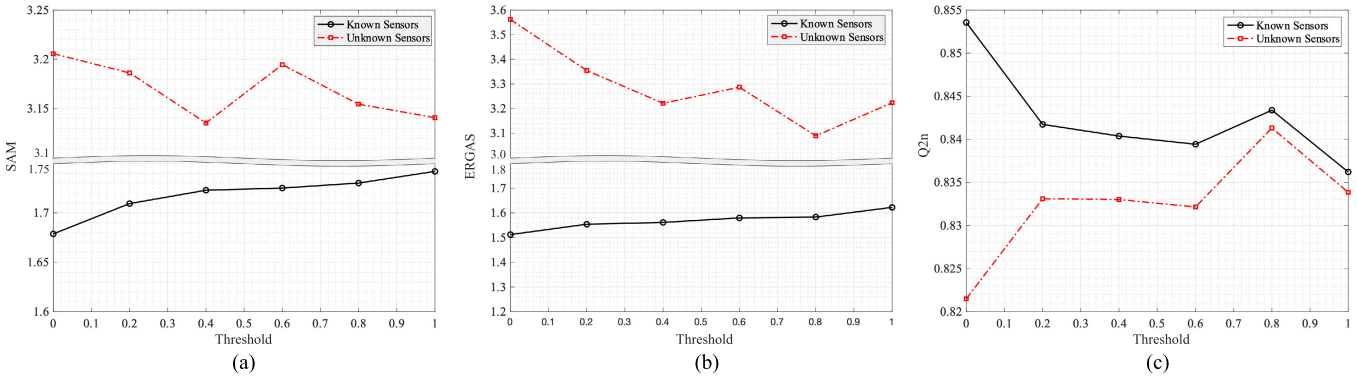


Fig. 3. Average (a) SAM, (b) ERGAS, and (c) Q2n performance on *known* and *unknown sensors* versus threshold.

every 2×10^4 iterations. The batch size is set to 8 and the patch size for random cropping is set to 64×64.

B. Ablation Studies

In this section, we conduct some experiments based on ArbRPN to validate the effectiveness of the proposed random anisotropic Gaussian-shaped MTF degradation and the GSD rescaling. Specifically, five models are listed in Table II. The meaning of each model is as follows.

1) "*Mix*": The model trained by the mixed (GF1, QB, and WV2) dataset.
2) "*Mix + Norm*": The model trained by the mixed dataset with data normalization.
3) "*Mix + Norm + RMTF*": The model trained by mixed dataset with data normalization and random isotropic Gaussian-shaped MTF degradation.
4) "*Mix + Norm + RMTF + An*": The model trained by mixed dataset with data normalization and random anisotropic Gaussian-shaped MTF degradation.
5) "*Mix + Norm + RMTF + An + SC*": The model trained by mixed dataset with data normalization, random anisotropic Gaussian-shaped MTF degradation, and GSD-rescaling.

1) Effectiveness of Data Normalization: Normalization is a common way to improve the generalization for many tasks. Therefore, we also validate its effectiveness for pansharpening. We use image-level z-score normalization, since we find this type of normalization is the best among all the normalization approaches, including image-level max-min normalization, band-level max-min normalization, and band-level z-score normalization. As shown in Table II, "Mix + Norm" shows advantages with respect to "Mix" for all the metrics on both the known and unknown sensors test cases.

2) Effectiveness of Random Isotropic Gaussian-Shaped MTF Degradation: Augmentation via random isotropic Gaussian-shaped MTF degradation can increase the diversity of samples applying different MTF degradations, thereby improving the robustness of deep models to tackle with MS images from different sensors. Comparing "Mix + Norm + RMTF" and "Mix + Norm," we can find that the former achieves better results than the latter on all the metrics (except for D_s). Therefore, the random isotropic Gaussian-shaped MTF module is also effective to improve average performance for *known sensors* and the generalization in the *unknown sensors* test case.

3) Effectiveness of Random Anisotropic Gaussian-Shaped MTF Degradation: Real MTFs of MS sensors can have an anisotropic shape [14]. Therefore, we apply random anisotropic Gaussian-shaped MTF degradations to training samples for better generalization. The approach is called "Mix + Norm + RMTF + AN" and it obtains the best results on all the RR metrics and the second-best D_λ for *known sensors*. For *unknown sensors*, it also achieves better results than "Mix + Norm + RMTF" on all the RR metrics and the D_λ index.

4) Effectiveness of GSD Rescaling: Sensors usually have different GSDs. Therefore, the GSD rescaling is introduce to augment data for training. To assess its effectiveness, we compare the "Mix + Norm + RMTF + AN + SC" approach (using GSD rescaling) and "Mix + Norm + RMTF + AN." As shown in Table II, the GSD rescaling obtains slightly worse results than the chain without GSD rescaling at RR for *known sensors*, but the clear advantages for the *unknown sensors* test case support the choice of adding the GSD rescaling module. Besides, the performance at RR is close to "Mix + Norm + RMTF + AN," but clearly better than this latter at FR, which demonstrates the effectiveness of the GSD rescaling

TABLE III

RESULTS OF MODELS TRAINED BY DIFFERENT DATA AUGMENTATION STRATEGIES. BEST RESULTS FOR EACH PANSHARPENING ARCHITECTURE ARE IN BOLDFACE. FOR PNN AND PANNET, ONLY SENSOR WITH FOUR-BAND MS IMAGES ARE TESTED SINCE THEY CANNOT DEAL WITH MS IMAGES WITH DIFFERENT NUMBERS OF BANDS

Methods	Average on *known sensors*, i.e., GF1, QB, and WV2							Average on *unknown sensors*, i.e., IK, WV3, SP7, PHR1A, and GE1.						
	RR Results				FR Results			RR Results				FR Results		
	SAM	ERGAS	Q2n	sCC	D_λ	D_S	HQNR	SAM	ERGAS	Q2n	sCC	D_λ	D_S	HQNR
PNN (GF1)	1.8935	1.4683	0.7917	0.9329	0.1022	0.0851	0.8191	5.4058	5.6684	0.6287	0.8238	0.1948	0.1322	0.6912
PNN (Mix)	1.4040	1.1971	0.8460	0.9548	0.0466	0.0895	0.8694	6.7483	5.6820	0.6039	0.8266	0.2092	0.1492	0.6722
PNN (ours)	**1.3337**	**1.1131**	**0.8659**	**0.9583**	**0.0279**	**0.0541**	**0.9208**	**5.3905**	**4.8346**	**0.6818**	**0.8612**	**0.1626**	**0.0997**	**0.7539**
PanNet (GF1)	1.4341	1.3747	0.8366	0.9346	0.0242	0.0637	0.9137	3.2274	4.6802	0.7331	0.8301	0.0452	0.1216	0.8371
PanNet (Mix)	1.2891	**1.1421**	0.8607	**0.9583**	0.0315	0.0438	0.9262	3.1414	4.3958	0.7434	0.8469	0.0585	0.1075	0.8382
PanNet (ours)	**1.2879**	1.1433	**0.8688**	0.9569	**0.0178**	**0.0378**	**0.9457**	**2.9061**	**3.7245**	**0.7977**	**0.8903**	**0.0362**	**0.0874**	**0.8792**
BIEDN (GF1)	3.2372	3.2275	0.7410	0.9044	0.1040	0.0856	0.8246	4.2905	5.1758	0.7235	0.8458	0.0915	0.1176	0.7976
BIEDN (Mix)	**1.8352**	1.6958	0.8356	**0.9697**	0.0424	0.0738	0.8923	3.5515	**3.7114**	0.8210	0.9193	0.0451	**0.0609**	**0.8968**
BIEDN (ours)	1.8539	**1.6852**	**0.8358**	0.9664	**0.0260**	**0.0633**	**0.9145**	**3.4138**	3.7125	**0.8289**	**0.9231**	**0.0333**	0.0773	0.8938
ArbRPN (GF1)	2.4909	3.0050	0.7344	0.9074	0.0533	0.0949	0.8567	4.1694	5.5836	0.6802	0.8109	0.0488	0.1159	0.8394
ArbRPN (Mix)	1.5111	1.3321	0.8580	0.9840	0.0460	**0.0865**	0.8766	3.9444	4.4400	0.7206	0.8743	0.0874	**0.1055**	0.8187
ArbRPN (ours)	**1.4908**	**1.3011**	**0.8628**	**0.9852**	**0.0254**	0.0986	**0.8808**	**3.0423**	**3.1953**	**0.8265**	**0.9331**	**0.0362**	0.1178	**0.8493**

module to improve the generalization with respect to spatial resolution.

5) Impacts of Threshold for GSD Rescaling: In our experiment using the GSD rescaling module, we found that the performance trade-off between *known* and *unknown sensors* is sensitive to the threshold exploited in this block (see Fig. 3). We can note more stable performance for *known sensors* varying the threshold than that of the *unknown sensors* test case. When the threshold is set to 0.8, the Q2n for *unknown sensors* achieves the best result, while the Q2n for *known sensors* is close to the best one. However, having a look at the results of SAM and ERGAS, we set the threshold to 0.2 to guarantee better performance for the *known sensors* test case while having satisfactory generalization as shown in Sections IV-B and IV-C.

6) Applicability to Other Architectures: To validate effectiveness of the proposed augmentation to other architectures, we consider well-known neural networks for pansharpening, i.e., PNN, PanNet, BIEDN, in addition to ArbRPN, trained by the GF1 dataset, the Mix (GF1, QB, and WV2) dataset, and "ours" (using Mix and the proposed data augmentation strategy). The results are shown in Table III. Regarding the comparison between the models trained by "GF1" and "Mix," we can find that "Mix" does not always improve the generalization of the model, e.g., PNN for *unknown sensors*. On other hand, the model trained by "Mix" with our augmentation strategy always obtains better results than "GF1."

C. Comparison With SOTA Methods

In this section, we will compare ArbRPN trained by the proposed data augmentation framework, denoted as ArbRPN+, with other SOTA methods for the *known sensors* test case. Afterward, we will compare ArbRPN+ for the other test case (i.e., *unknown sensors*).

1) Benchmark: To show the effectiveness of ArbRPN+, we compare it with four SOTA DL methods for pansharpening. Four metrics (i.e., SAM, ERGAS, Q2n, sCC) are used to evaluate fused results at RR. The ideal value is 0 for SAM and ERGAS, and 1 for Q2n and sCC. Three without reference metrics, D_λ, D_s and hybrid quality with non-reference (HQNR), are used to evaluate results at FR. The ideal value is 0 for D_λ and D_s, and 1 for HQNR. Standard implementations have been considered of these quality indexes [27], [28]. Specifically, the adopted benchmark is as follows.

1) *EXP*: A 23-taps based interpolation applied for pansharpening [31].
2) *PNN*: CNNs for pansharpening [15].
3) *PanNet*: Pansharpening network with high-pass filtered MS and PAN images [19].
4) *BIEDN*: A band independent encoder–decoder network for pansharpening [22].
5) *ArbRPN*: A pansharpening network for MS images with arbitrary numbers of bands [2].

2) Results on Known Sensors: The numerical results for this test case are reported in Table IV.

Results at RR: According to the results shown in Table IV, on the one hand, we can find that ArbRPN+ achieves just slightly worse results than ArbRPN. This is expected because ArbRPN is a model dedicated to a specific sensor, which can be viewed as the upper bound of ArbRPN+ for the given specific sensor. On the other hand, ArbRPN+ outperforms all the methods except ArbRPN, showing that training a model for various sensors is feasible. To sum up, we have that ArbRPN+ achieves results close to its upper bound (ArbRPN) and outperforms the other SOTA methods at RR for the *known sensors* test case demonstrating the ability of the proposed data augmentation framework.

A visual comparison of an RR sample is shown in Fig. 4(a). According to the RGB results, we can hardly say something looking at 8-bits RGB images. On the other hand, we can observe the residual images that clearly point out the advantages in using ArbRPN and ArbRPN+ (fewer errors than the compared approaches).

Results at FR: The results at FR in Table IV show a sensor-based variability. However, looking at the overall results by averaging for sensors, we can find that ArbRPN+ achieves the best results followed by ArbRPN. There are two reasons to justify the inversion in performance with respect to the RR test cases. The first reason is due to the fact that ArbRPN+ can easily generalize with respect to the spatial degradation model. Instead, ArbRPN relies upon fixed filters that can diverge with respect to the real (and unknown) spatial models under the particular sensor. The second point is the generalization ability

TABLE IV

AVERAGE RESULTS ON DATA FROM *Known Sensors*. THE BEST AND THE SECOND-BEST RESULTS ARE IN BOLDFACE AND UNDERLINED, RESPECTIVELY

Methods	GF1							QB						
	RR Results				FR Results			RR Results				FR Results		
	SAM	ERGAS	Q2N	sCC	D_λ	D_S	HQNR	SAM	ERGAS	Q2N	sCC	D_λ	D_S	HQNR
EXP	1.5506	1.5667	0.8086	0.9057	**0.0110**	0.0676	0.9222	1.8491	1.8315	0.6813	0.8622	0.0337	0.0962	0.8730
PNN	1.2928	1.2152	0.8941	0.9499	0.0280	0.0947	0.8798	1.1035	0.8830	0.8633	0.9768	0.0434	0.0623	0.9000
PanNet	1.2864	1.2190	0.8892	0.9511	0.0171	0.0524	0.9316	1.0440	0.8745	0.8711	0.9779	0.0303	0.0557	0.9176
BIEDN	1.1934	1.1154	0.9122	0.9582	0.0165	**0.0497**	**0.9346**	0.9331	0.7195	0.8805	0.9850	0.0180	0.0414	0.9420
ArbRPN	**0.7435**	**0.6211**	**0.9586**	**0.9878**	0.0175	0.0936	0.8900	**0.7313**	**0.5975**	**0.9014**	**0.9908**	0.0208	**0.0374**	**0.9433**
ArbRPN+	0.8416	0.6944	0.9522	0.9844	**0.0110**	0.0776	0.9132	0.8329	0.6288	0.8938	0.9881	**0.0144**	0.0658	0.9207
	WV2							AVERAGE						
	SAM	ERGAS	Q2N	sCC	D_λ	D_S	HQNR	SAM	ERGAS	Q2N	sCC	D_λ	D_S	HQNR
EXP	4.8346	7.3138	0.4650	0.7448	0.1031	**0.1036**	0.8048	2.7448	3.5707	0.6517	0.8367	0.0493	0.0891	0.8667
PNN	3.3222	3.1276	0.7258	0.9715	0.2009	0.1767	0.6859	1.9062	1.7419	0.8277	0.9661	0.0907	0.1112	0.8219
PanNet	3.4067	3.2344	0.7258	0.9693	0.1460	0.1541	0.7395	1.9124	1.7760	0.8287	0.9661	0.0644	**0.0874**	0.8629
BIEDN	3.2687	3.1168	0.7227	0.9716	0.1267	0.1872	0.7294	1.7984	1.6506	0.8385	0.9716	0.0537	0.0928	0.8687
ArbRPN	**2.6868**	**2.4047**	0.7418	**0.9859**	0.0832	0.1432	0.7941	**1.3872**	**1.2078**	**0.8673**	**0.9882**	0.0405	0.0914	0.8758
ArbRPN+	2.7980	2.5801	**0.7425**	0.9830	**0.0509**	0.1535	**0.8084**	1.4908	1.3011	0.8628	0.9852	**0.0254**	0.0986	**0.8808**

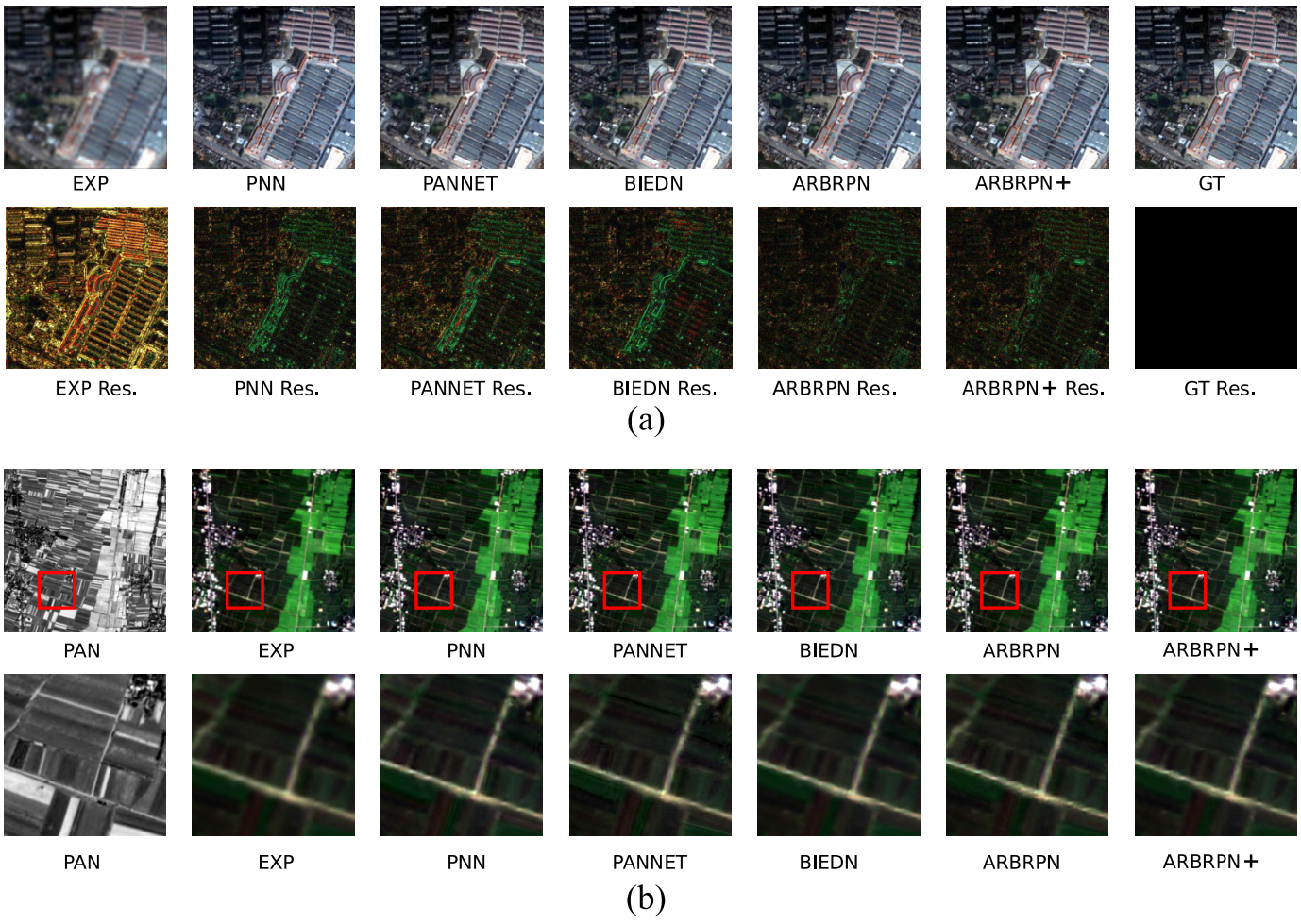


Fig. 4. Visual comparison for *known sensors*. (a) QB RR sample. "Res." refers to the absolute residual image between the GT and the image obtained by the corresponding method. (b) GF1 FR sample. The bottom row shows close-ups, i.e., the red rectangle area in the corresponding image in the top row.

of ArbRPN+ with respect to the changes of GSDs. Indeed, the assumption of invariance among scales is not generally valid for the training of neural networks pushing toward the recent development of unsupervised learning strategies. Thus, ArbRPN+ can address this problem providing a great generalization ability of the network in the GSD direction getting the same goal as unsupervised learning but working at RRs following the same principles as done for the FR quality assessment using extrapolation [32], [33].

A visual comparison of FR samples is shown in Fig. 4(b). Again, for the other SOTA results, the visual comparison is hard because of both the absence of a reference image and of the comparison of three bands with 8-bits radiometric resolution images. The differences among the compared

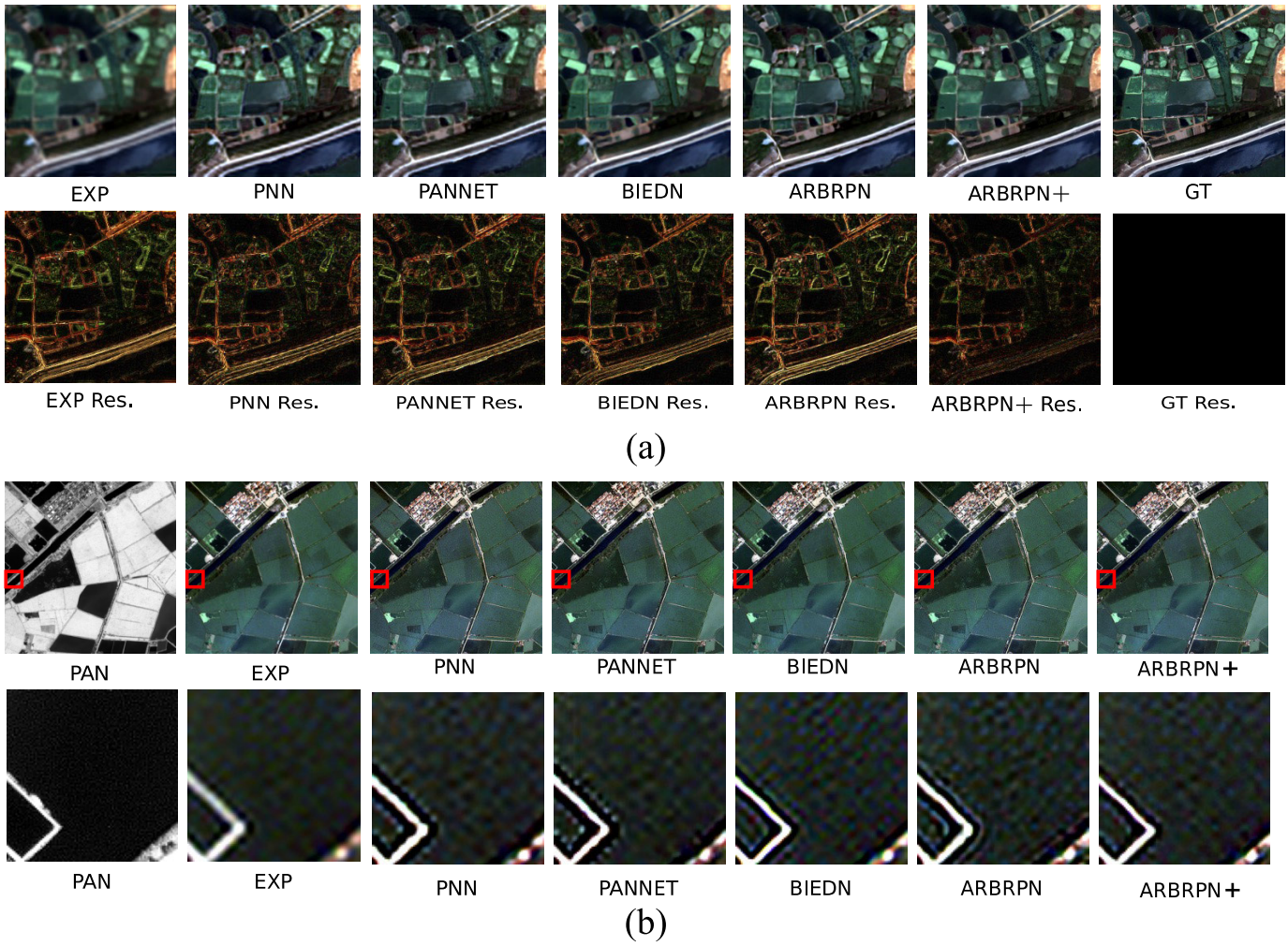


Fig. 5. Visual comparison for the *unknown sensors* test case. (a) IK RR sample. "Res." refers to the absolute residual image between the GT and the image obtained by the corresponding method. (b) IK FR sample. The bottom row shows close-ups, i.e., the red rectangle area in the corresponding image in the top row.

TABLE V

AVERAGE RESULTS FOR *Unknown Sensors*. THE BEST AND THE SECOND-BEST RESULTS ARE IN BOLDFACE AND UNDERLINED, RESPECTIVELY

Methods	RR Results				FR Results		
	SAM	ERGAS	Q2n	sCC	D_λ	D_S	HQNR
EXP	4.0924	5.7735	0.6382	0.7555	0.0407	**0.1113**	**0.8521**
PNN	5.4058	5.6684	0.6287	0.8238	0.1948	0.1322	0.6912
PanNet	3.2274	4.6802	0.7331	0.8301	0.0452	0.1216	0.8371
BIEDN	4.2905	5.1758	0.7235	0.8458	0.0915	0.1176	0.7976
ArbRPN	4.1694	5.5836	0.6802	0.8109	0.0488	0.1159	0.8394
ArbRPN+	**3.0423**	**3.1953**	**0.8265**	**0.9331**	**0.0362**	0.1178	0.8493

approaches will become more clear for the more challenging *unknown sensors* test case [see Fig. 5(b)].

3) Generalization on Unknown Sensors: To validate the generalization of the proposed data augmentation framework for *unknown sensors*, we compare the SOTA methods trained for one known sensor, specifically, the models trained by GF1 data in Table IV, and ArbRPN+, which is also the same model in Table IV. The average results are reported in Table V.

Results at RR: According to the results at RR, we can find that the performance gap between the DL methods and EXP become smaller than that on *known sensors*. PNN achieves even worse results than EXP for all the metrics except for sCC. ArbRPN and BIEDN achieve worse SAM results than EXP as well. Only PanNet and ArbRPN+ obtain better results than EXP for all the metrics. Moreover, ArbRPN+ shows a big improvement compared with ArbRPN and all the other methods, which indicates the effectiveness of the proposed data augmentation mechanism.

Fig. 5(a) shows a visual comparison for an IK RR sample. From the RGB images, we can find that EXP obtain more blurred results than the other methods. The results of PNN have some unexpected textures in the arable land. Similar to the visual comparison for known sensors, we visualize the residual images between GT and the result obtained by the corresponding method. It is straightforward that ArbRPN+ achieves the best results showing fewer errors than all the other fused products.

Results at FR: As shown in Table V, excluding the EXP approach that produces no spatial enhancement, ArbRPN+ achieves the best results on FR test cases. A visual comparison is provided in Fig. 5(b), according to the close-ups, we can see clear spatial distortions, i.e., the ringing artifacts close to the edges, for PNN, PANNET, BIEDN, and ArbRPN. Conversely, ArbRPN+ has fewer spatial distortions than these methods, thus showing better performance for ArbRPN+.

V. Conclusion

In this article, we studied the generalization of PNNs from the perspective of data augmentation. According to the MTF approximation using gains at Nyquist frequency, a random anisotropic Gaussian-shaped MTF degradation has been proposed to increase the diversity in degrading MS images for training neural networks. Furthermore, a GSD rescaling module has been introduced to both improving the networks generalization and the performance at FR increasing the generalization with respect to spatial resolution. The ablation study demonstrated the effectiveness of these augmentation approaches. Experimental results showed that a network trained by the proposed data augmentation framework gets better generalizations with respect to both sensor and resolution. It is worth emphasizing that the proposed approach is almost a free-lunch to get better generalization retaining the same number of iterations for training of the original architecture where it has been applied.

References

[1] C. Lanaras, J. Bioucas-Dias, S. Galliani, E. Baltsavias, and K. Schindler, "Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network," *ISPRS J. Photogram. Remote Sens.*, vol. 146, pp. 305–319, Dec. 2018.

[2] L. Chen, Z. Lai, G. Vivone, G. Jeon, J. Chanussot, and X. Yang, "ArbRPN: A bidirectional recurrent pansharpening network for multispectral images with arbitrary numbers of bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5406418.

[3] 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.

[4] S. Luo, S. Zhou, Y. Feng, and J. Xie, "Pansharpening via unsupervised convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4295–4310, 2020.

[5] Y. Qu, R. K. Baghbaderani, H. Qi, and C. Kwan, "Unsupervised pansharpening based on self-attention mechanism," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3192–3208, Apr. 2021.

[6] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion," *Inf. Fusion*, vol. 62, pp. 110–120, Oct. 2020.

[7] H. Zhou, Q. Liu, and Y. Wang, "PGMAN: An unsupervised generative multiadversarial network for pansharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6316–6327, 2021.

[8] H. Zhou, Q. Liu, D. Weng, and Y. Wang, "Unsupervised cycle-consistent generative adversarial networks for Pan sharpening," *IEEE Trans. Geosci. Remote. Sens.*, vol. 60, 2022, Art. no. 5408814.

[9] M. Ciotola, S. Vitale, A. Mazza, G. Poggi, and G. Scarpa, "Pansharpening by convolutional neural networks in the full resolution framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5408717.

[10] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[11] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.

[12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[13] J. Yoo, N. Ahn, and K.-A. Sohn, "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8375–8384.

[14] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "MTF-tailored multiscale fusion of high-resolution MS and PAN imagery," *Photogramm. Eng. Remote Sens.*, vol. 72, no. 5, pp. 591–596, May 2006.

[15] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sens.*, vol. 8, no. 7, pp. 594–615, Jul. 2016.

[16] 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. 2015.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Z. Lai, L. Chen, Z. Liu, and X. Yang, "Gradient guided pyramidal convolution residual network with interactive connections for pan-sharpening," *Int. J. Remote Sens.*, vol. 43, pp. 5572–5602, Aug. 2022.

[22] C. Liu et al., "Band-independent encoder–decoder network for pan-sharpening of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 5208–5223, Jul. 2020.

[23] F. Ozcelik, U. Alganci, E. Sertel, and G. Unal, "Rethinking CNN-based pansharpening: Guided colorization of panchromatic images via GANs," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3486–3501, Apr. 2021.

[24] D. Lei, H. Chen, L. Zhang, and W. Li, "NLRNet: An efficient nonlocal attention ResNet for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5401113.

[25] 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.

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

[27] G. Vivone, M. Dalla Mura, A. Garzelli, and F. Pacifici, "A benchmarking protocol for pansharpening: Dataset, preprocessing, and quality assessment," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6102–6118, 2021.

[28] G. Vivone et al., "A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 53–81, Mar. 2021.

[29] X. Meng et al., "A large-scale benchmark data set for evaluating pansharpening performance: Overview and implementation," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 18–52, Mar. 2021.

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

[31] B. Aiazzi, L. Alparone, S. Baronti, and A. Garzelli, "Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2300–2312, Oct. 2002.

[32] R. Carlà, L. Santurri, B. Aiazzi, and S. Baronti, "Full-scale assessment of pansharpening through polynomial fitting of multiscale measurements," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6344–6355, Dec. 2015.

[33] G. Vivone, R. Restaino, and J. Chanussot, "A Bayesian procedure for full-resolution quality assessment of pansharpened products," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4820–4834, Aug. 2018.

Lihui Chen (Member, IEEE) received the B.Sc. degree in electronics and information engineering and the Ph.D. degree in information and communication engineering from Sichuan University, Chengdu, China, in 2018 and 2022, respectively.

He is currently an Assistant Researcher with Chongqing University, Chongqing, China. He visited the National Research Council, Institute of Methodologies for Environmental Analysis, CNR-IMAA, Tito, Italy, from August 2021 to August 2022. His research interests include image processing, remote sensing, and deep learning.

Gemine Vivone (Senior Member, IEEE) received the B.Sc. (summa cum laude), M.Sc. (summa cum laude), and Ph.D. degrees in information engineering from the University of Salerno, Fisciano, Italy, in 2008, 2011, and 2014, respectively.

In 2019, he was an Assistant Professor with the University of Salerno. In 2014, he joined the North Atlantic Treaty Organization (NATO), Science and Technology Organization (STO), Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, as a Scientist. He was a Visiting Professor with the Grenoble Institute of Technology (INPG), Grenoble, France. He is currently a Researcher with the National Research Council, Rome, Italy. His main research interests include statistical signal processing, detection of remotely sensed images, data fusion, and tracking algorithms.

Dr. Vivone received the IEEE-GRSS Early Career Award in 2021, the Symposium Best Paper Award at the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) in 2015, and the Best Reviewer Award of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2017. He is the Co-Chair for the IEEE Image Analysis and Data Fusion Technical Committee. He is currently an Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL). He is an Editorial Board Member of *MDPI Remote Sensing*, *MDPI Sensors*, and *MDPI Encyclopedia*. He served as a guest associate editor for several special issues.

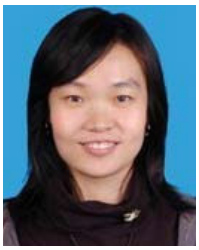
Zihao Nie received the B.Sc. degree in electronics and information engineering from Sichuan University, Chengdu, China, in 2021, where he is currently pursuing the M.Sc. degree in pattern recognition with the College of Electronics and Information Engineering.

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

Jocelyn Chanussot (Fellow, IEEE) received the M.Sc. degree in electrical engineering from the Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree from the Université de Savoie, Annecy, France, in 1998.

Since 1999, he has been with Grenoble INP, where he is currently a Professor of signal and image processing. He has been a Visiting Scholar with Stanford University, Stanford, CA, USA; the KTH Royal Institute of Technology, Stockholm, Sweden; and the National University of Singapore (NUS), Singapore. Since 2013, he has also been an Adjunct Professor with the University of Iceland, Reykjavik, Iceland. From 2015 to 2017, he was a Visiting Professor with the University of California at Los Angeles (UCLA), Los Angeles, CA, USA. He holds the AXA Chair in remote sensing and is an Adjunct professor at the Chinese Academy of Sciences, Aerospace Information research Institute, Beijing, China. He has coauthored over 250 articles in international journals, gathering 31 500+ citations, with H-index = 78. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning, and artificial intelligence.

Dr. Chanussot is an ELLIS Fellow, is a fellow of the Asia-Pacific Artificial Intelligence Association, and was a member of the Institut Universitaire de France from 2012 to 2017. He is the founding President of the IEEE Geoscience and Remote Sensing French Chapter from 2007 to 2010, which received the 2010 IEEE GRS-S Chapter Excellence Award. He has received multiple outstanding paper awards. He was the Vice President of the IEEE Geoscience and Remote Sensing Society, in charge of meetings and symposia from 2017 to 2019. He was the General Chair for the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS). He was the Chair from 2009 to 2011 and the Co-Chair from 2005 to 2008 for the GRS Data Fusion Technical Committee. He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society from 2006 to 2008 and the Program Chair for the IEEE International Workshop on Machine Learning for Signal Processing in 2009. He is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the PROCEEDINGS OF THE IEEE. He was the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING from 2011 to 2015. In 2014, he served as a Guest Editor for the *IEEE Signal Processing Magazine*. He has been a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters) since 2018.



Xiaomin Yang (Member, IEEE) received the B.S. and Ph.D. degrees in communication and information system from Sichuan University, Chengdu, China, in 2002 and 2007, respectively.

She was a Post-Doctoral Researcher with The University of Adelaide, Adelaide, SA, Australia, for one year. She is currently a Professor with the College of Electronics and Information Engineering, Sichuan University. Her research interests include image processing and pattern recognition.