Detail Injection-Based Deep Convolutional Neural Networks for Pansharpening

Liang-Jian Deng®, Member, IEEE, Gemine Vivone®, Senior Member, IEEE, Cheng Jin®, and Jocelyn Chanussot®, Fellow, IEEE

Abstract—The fusion of high spatial resolution panchromatic (PAN) data with simultaneously acquired multispectral (MS) data with the lower spatial resolution is a hot topic, which is often called pansharpening. In this article, we exploit the combination of machine learning techniques and fusion schemes introduced to address the pansharpening problem. In particular, deep convolutional neural networks (DCNNs) are proposed to solve this issue. The latter is combined first with the traditional component substitution and multiresolution analysis fusion schemes in order to estimate the nonlinear injection models that rule the combination of the upsampled low-resolution MS image with the extracted details exploiting the two philosophies. Furthermore, inspired by these two approaches, we also developed another DCNN for pansharpening. This is fed by the direct difference between the PAN image and the upsampled low-resolution MS image. Extensive experiments conducted both at reduced and full resolutions demonstrate that this latter convolutional neural network outperforms both the other detail injection-based proposals and several state-of-the-art pansharpening methods.

Index Terms—Component substitution (CS), deep convolutional neural network (DCNN), image fusion, multiresolution analysis (MRA), pansharpening, remote sensing.

I. INTRODUCTION

PANSHARPENING has become a fundamental problem in remote sensing image processing since it can fuse a textithigh spatial resolution panchromatic (PAN) image and a low spatial resolution multispectral (MS) image in order to obtain an MS image with the highest (PAN) spatial resolution. PAN and MS images are quite common in the field of remote sensing imaging, and they are usually simultaneously acquired by sensors mounted on many satellites, such as IKONOS, WorldView-2, and WorldView-3. Pansharpening has attracted the interest of the scientific community. This is justified by the contest launched by the Data Fusion Committee of the IEEE Geoscience and Remote Sensing Society in 2006 [3], [4] and many recently published review articles [5], [6]. Furthermore, pansharpened products have attracted the interest of some commercial companies, e.g., Google Earth, and pansharpening has been exploited as preliminary a step for several image processing tasks, e.g., change detection [7], [8].

Most of the pansharpening works can be divided into four categories, i.e., component substitution (CS) methods, multiresolution analysis (MRA) approaches, variational optimization-based (VO) techniques, and machine learning (ML) approaches.

CS and MRA approaches play an important role in the community of pansharpening. They have shown promising performance with a balanced computational burden. The CS-based methods rely on the concept of the projection of the MS image into a new domain, where the spatial information can be easily separated into a component, usually called the intensity component. Then, the (possibly equalized) PAN image can be substituted with the intensity component. The sharpened version of the MS image is obtained due to the inverse projection bringing the data to the original MS domain. CS-based methods can generate outcomes with high spatial fidelity paid by a usually greater spectral distortion. Some powerful instances of methods belonging to this category are the band-dependent spatial-detail (BDSD) with local parameter estimation [9], the robust (BDSD-PC) method [10], the partial replacement adaptive component substitution (PRACS) [11], and Gram–Schmidt (GS) spectral sharpening [12].

MRA-based approaches inject spatial details extracted from the PAN image through an MRA framework into the MS image in order to get the high spatial resolution MS image. MRA-based products preserve spectral information but can suffer from spatial distortion. The examples of methods into this class are the smoothing filter-based intensity modulation (SFIM) [13], the additive wavelet luminance proportional (AWLP) [14], the “à-trous” wavelet transform [15], the Laplacian pyramid (LP) [16], the generalized Laplacian pyramid (GLP) [17], [18], the GLP with robust regression [19], and the GLP with full-scale regression (GLP-Reg) [20].

Recently, VO approaches have shown competitive ability in addressing the pansharpening issue. Techniques belonging to this class include the Bayesian methods [21]–[23],
variational approaches [24]–[36], and compressed sensing techniques [37]–[39]. Despite their formal mathematical elegance, VO approaches provide only incremental performance improvements with respect to the state-of-the-art of CS and MRA methods; such improvement comes at the cost of high computational burden and presence of many parameters to be tuned explaining why CS and MRA are nowadays commonly advocated both for benchmarking and practical uses.

With the tremendous improvements of hardware, convolutional neural networks (CNNs) have recently become a powerful tool to deal with pansharpening and its related applications (see [1], [2], [40]–[54]). The CNN-based methods depend on the large-scale data set training to learn a nonlinear functional mapping between the low spatial resolution MS images and the high spatial resolution MS images. After the training phase, it is easy to predict/compute the pansharpened image by the learned nonlinear mapping. Masi et al. [41] first proposed a simple and effective CNN architecture with three layers called pansharpening neural network (PNN). This architecture is mainly based on a previous CNN architecture for single image super-resolution [55] and yields state-of-the-art pansharpening outcomes. Liu et al. [52] presented a good way to inject the high-pass details of the PAN image into the upsampled MS image, even by exploiting classical injection gains. This way is a bit like the scheme of traditional CS and MRA methods, but the extraction of the high-pass details is not in agreement with the classical procedures performed by CS and MRA approaches. Yang et al. [1] proposed a deeper network architecture than PNN, which is called PanNet. The PanNet architecture incorporates domain-specific knowledge and mainly focuses on two important issues, i.e., spectral and spatial preservations, obtaining state-of-the-art results. Furthermore, due to the use of high-pass filtering, the given architecture also shows the relevant ability of network generalization. He et al. [2] proposed a detail-injection-based convolutional neural network (DiCNN). In particular, the authors developed two detail injection-based architectures, i.e., DiCNN1, whose detail injection depends on both MS and PAN images, and DiCNN2, whose detailed injection depends only on PAN images. DiCNN2 is designed to alleviate the computational burden; instead, DiCNN1 is more oriented to high quantitative performance getting state-of-the-art results. However, there is still room for improvement focusing on aspects as network complexity, training time, robustness, and so forth.

In this article, we propose deep CNNs to address the pansharpening problem, even accounting for fusion schemes proposed in the literature. In particular, we focus our attention on traditional CS and MRA frameworks. The details are extracted using these two philosophies. Instead, the nonlinear injection model is estimated through CNNs. These approaches are here named CS-Net and MRA-Net, respectively. Inspired by these solutions, we further investigate this idea feeding the network with details directly extracted by differencing the single PAN image with each MS band. This solution allows us to avoid compromising the spatial information with a preprocessing step using detailed extraction techniques proposed in classical pansharpening approaches, letting the CNN spectrally adjust the extracted details (e.g., the details are clearly biased) through the estimation of the nonlinear and local injection model. This approach will be called Fusion-Net from hereon. The proposed approaches are tested on several data sets acquired by the WorldView-2, WorldView-3, GaoFen (GF)-2, and QuickBird (QB) data sets. The experimental analysis is conducted both at reduced and full resolutions. The benchmark consists of state-of-the-art CS and MRA approaches and ML methods for pansharpening. The proposed Fusion-Net method clearly shows state-of-the-art performance outperforming the methods in the adopted benchmark both quantitatively and qualitatively. Finally, discussions about network complexity, training time, convergence, and robustness are provided to the readers for all the compared CNN approaches.

In summary, the main contributions of this work are as follows.

1) Two physically justified CNNs (i.e., CS-Net and MRA-Net) have been proposed deriving them from the traditional CS and MRA frameworks.

2) Inspired by CS-Net and MRA-Net, the Fusion-Net has also been proposed to reach state-of-the-art performance (see the comparison among the high-performance CNNs in Fig. 1 for a WorldView-3 data set). Moreover, the Fusion-Net has a simple architecture with fewer network parameters, thus resulting in more effective than some previously developed network architectures for pansharpening.
3) A broad experimental analysis has been provided based on several data sets. The performance is assessed both at reduced and full resolutions. The numerical outcomes are also corroborated by a qualitative analysis. Finally, a deep discussion on the network generalization, convergence property, computational time, and robustness on large data sets has been provided to the readers for all the considered CNN approaches.

This article is organized as follows. The related works and motivations are introduced in Section II. The proposed three network architectures will be detailed in Section III. Section IV is devoted to the description of the experimental results and the related discussions. Finally, conclusions are drawn in Section V.

II. RELATED WORKS AND MOTIVATIONS

The proposed network is initially inspired by two traditional pansharpening frameworks, i.e., CS and MRA. Therefore, we will first introduce them in this section, and then, we will move toward the motivations under the choice of the proposed network architectures.

A. CS

The general fusion equation for CS-based methods is as follows:

\[
\hat{\mathbf{M}}_i = \mathbf{M} + g_i (\mathbf{P} - \mathbf{I}_L), \quad i = 1, 2, \ldots, B
\]  

where \(\hat{\mathbf{M}}_i\) is the \(i\)th band of the high spatial resolution MS image, \(\mathbf{M} \in \mathbb{R}^{M \times N}\) is the \(i\)th band of the upsampled version of the low spatial resolution MS image, \(g_i \in \mathbb{R}\) is the \(i\)th injection coefficient (a real number for global phases) that controls the injection of the extracted details, \(\mathbf{P} \in \mathbb{R}^{M \times N}\) represents the PAN image, and \(\mathbf{I}_L \in \mathbb{R}^{M \times N}\) is the intensity component, generally defined as \(\mathbf{I}_L = \sum_{i=1}^{B} \omega_i \mathbf{M}_i\), where \(\omega_i \in \mathbb{R}\) is the \(i\)th weight. Many CS-based pansharpening algorithms rely upon (1), just changing the ways to estimate the injection coefficients \(g_i\) and the weights \(\omega_i\) (see [9], [11]–[13]). Equation (1) could be further rewritten in the following multiband form:

\[
\hat{\mathbf{M}} = \mathbf{M} + \mathbf{g} \odot (\mathbf{P} - \mathbf{I}_L) \quad (2)
\]

where \(\hat{\mathbf{M}} \in \mathbb{R}^{M \times N \times B}\) and \(\mathbf{M} \in \mathbb{R}^{M \times N \times B}\) are obtained by stacking the bands \(\mathbf{M}_i, i = 1, 2, \ldots, B\), and \(\mathbf{M}_i, i = 1, 2, \ldots, B\), respectively, \(\mathbf{P} \in \mathbb{R}^{M \times N \times B}\) and \(\mathbf{I}_L \in \mathbb{R}^{M \times N \times B}\) are yielded by duplicating along the spectral dimension the PAN image, \(\mathbf{P}\), and the intensity component, \(\mathbf{I}_L\), respectively. \(\mathbf{g} = (g_1, g_2, \ldots, g_B)^T \in \mathbb{R}^B\) is a vector of coefficients \(g_i\) as in (1), and \(\odot\) is an operator indicating that the \(i\)th element of \(\mathbf{g}\) multiplies the \(i\)th spectral band of \(\mathbf{P} - \mathbf{I}_L\).

B. MRA

Similar to the CS-based method, the MRA-based method follows the following equation:

\[
\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{g} \odot (\mathbf{P} - \mathbf{P}_L) \quad (3)
\]

where \(\mathbf{M}, \mathbf{M}, \mathbf{P}, \mathbf{g}\), and the operator \(\circ\) have the same definitions as in (2). Different from \(\mathbf{P}_L\) in (2), \(\mathbf{P}_L \in \mathbb{R}^{M \times N \times B}\) is yielded by duplicating along the spectral dimension the low-pass spatial resolution version of the PAN image, \(\mathbf{P}\). By differencing \(\mathbf{P} - \mathbf{P}_L\), i.e., \(\mathbf{P} - \mathbf{P}_L\), the PAN spatial details can be extracted. Classical MRA approaches differ from each other in the way to extract PAN details and how to estimate the injection coefficient \(\mathbf{g}\) in (3) (see [14], [15], [17]).

C. Motivations

The CS and MRA approaches have achieved promising performance in the field of pansharpening. However, a big limitation for both the classes is the common assumption of using linear injection models, which does not generally hold having a look at the relative spectral responses of sensors usually exploited for pansharpening (e.g., it is easy to notice the overlaps among the MS spectral responses). This consideration has motivated us to avoid linear injection models developing nonlinear approaches, aiming to replace the detailed injection phases in both CS and MRA methods. Deep convolutional neural networks (DCNNs) can easily manage this nonlinear mapping task due to the fact that they are able to reproduce strong nonlinearities in the data. Thus, they represent the best solution for the problem at hand. In particular, we still follow the general classical framework based on two phases: 1) detail extraction and 2) detail injection into the original MS image. However, we address the issue of nonlinear and local estimation of injection coefficients leveraging on DCNNs. Thus, in what follows, we will present the three proposed solutions based on different DCNN architectures for pansharpening (i.e., CS-Net, MRA-Net, and Fusion-Net).

III. PROPOSED NETWORK ARCHITECTURES

This section is devoted to the presentation of the DCNNs proposed in this work. We will present first the two CS- and MRA-based networks. Afterward, the Fusion-Net will be detailed.

A. CS-Net

Let us recall (2), in which the pansharpened product \(\hat{\mathbf{M}}\) is equal to the sum of the upsampled MS image \(\mathbf{M}\) and the injected details \(\mathbf{g} \odot (\mathbf{P} - \mathbf{I}_L)\). In this equation, the upsampled MS image \(\mathbf{M}\) holds the spatial information at low resolution, and \((\mathbf{P} - \mathbf{P}_L)\) provides the high frequency details, injected through \(\mathbf{g}\).

Equation (2) requires the estimation of the injection coefficients \(\mathbf{g}\). Instead, we ignore that the injection coefficients considering the pansharpened image, \(\mathbf{M}\), consist of the upsampled MS image \(\hat{\mathbf{M}}\) plus the details coming from the nonlinear mapping provided by a DCNN feeding it with \((\mathbf{P} - \mathbf{P}_L)\). In summary, the CS-Net can be summarized as follows:

\[
\tilde{\mathbf{M}} = \mathbf{M} + f_{\Theta_{CS}}(\mathbf{P} - \mathbf{P}_L) \quad (4)
\]

where \(f_{\Theta_{CS}}\) is the nonlinear mapping with the network parameter \(\Theta_{CS}\) that could be learned from a large-scale training.
Fig. 2. Architecture of the CS-Net. The upsampling is performed using a polynomial kernel with 23 coefficients [17]. For “NetWork,” please refer to Section III-D.

Fig. 3. Architecture of the MRA-Net. The upsampling is performed using a polynomial kernel with 23 coefficients [17]. For “NetWork,” please refer to Section III-D.

data set. Several solutions to get the weights $\omega_i, i = 1, \ldots , B$ for $I_L$ are given by the pansharpening literature, from constant to band-dependent (estimated) weights. From our broad experimental analysis, comparable results can be obtained by the CS-Net using these different intensity components.

Starting from (4), it is easy to build the corresponding network architecture (see Fig. 2). In particular, the final loss function for the CS-Net can be defined under the metric of mean squared error (MSE) computed on training examples. Hence, we have

$$\text{Loss}(\Theta_{CS}) = \frac{1}{n} \sum_{k=1}^{n} \left\| \tilde{MS}_k + f_{\Theta_{CS}}(P_{D} - P_{D,L(k)}) - GT_k \right\|_F^2$$

(5)

where $n$ represents the number of training examples, $\| \cdot \|_F$ is the Frobenius norm, and $GT_k$ is the $k$th example extracted from the ground-truth (GT) image. By minimizing the loss function (5), the network $f_{\Theta_{CS}}$ will be enforced to automatically learn an optimal mapping with parameters $\Theta_{CS}$. Thus, the fusion can be completed by summing the weighted spatial details to the upsampled MS image following (4).

B. MRA-Net

Similar to the analysis of the CS-Net, we derive the architecture of MRA-Net. The pansharpened image $\hat{MS}$ in (3) consists of the upsampled MS image $\tilde{MS}$ plus the injected details $g(P_D - P_{D,L})$. Again, we ignore the injection coefficients $g$ by imposing a nonlinear mapping function estimated through a DCNN fed by $(P_D - P_{D,L})$. Therefore, the MRA-Net can be summarized as follows:

$$\hat{MS} = \tilde{MS} + f_{\Theta_{MRA}}(P_D - P_{D,L})$$

(6)

where $f_{\Theta_{MRA}}$ is the nonlinear mapping function with network parameters $\Theta_{MRA}$. Several solutions to get $P_{D,L}$ from $P$ are given by the pansharpening literature, from average to Gaussian filters. Again, from our broad experimental analysis, comparable results can be obtained by the MRA-Net using these different ways to spatially filter the PAN image.

Using (6), it is easy to design the network architecture of the MRA-Net (see Fig. 3). In particular, the loss function of
the MRA-Net is defined as follows:

$$\text{Loss}(\Theta_{\text{MRA}}) = \frac{1}{n} \sum_{k=1}^{n} \frac{\|\text{MS}^{(k)} + f_{\Theta_{\text{MRA}}}(\text{PD}^{(k)} - \text{PD}_{L}^{(k)}) - \text{GT}^{(k)}\|_F^2}{\sum_{k=1}^{n} \|\text{MS}^{(k)}\|_F^2}$$

(7)

where the definitions of symbols are the same as that of (5).

C. Fusion-Net

CS-Net and MRA-Net have been developed starting from the two classical fusion schemes related to CS and MRA. Thus, they have a solid and physical justification rooted in the pansharpening literature. However, in order to extract the details, preliminary assumptions should be done either on the shape of the spatial filters (for MRA-Net) or on the spectral model ruling the projection of the MS image into the PAN domain (for CS-Net). Errors in this phase can have a great impact on the outcomes reducing the performance of the proposed approaches. Thus, aiming of having a detail-based architecture, but avoiding the above-mentioned issue, the solution of subtracting the duplicated version of the PAN image, $\text{PD}$, with the upsampled MS image, $\text{MS}$, is advisable. This has also the advantage to alleviate the computation burden of the approach avoiding to calculate $\text{PD}$ or $\text{PD}_{L}$. The limitation of this solution is instead related to the strong spectral distortion introduced in the extracted details (e.g., biased details) that can be easily compensated by the network during its training phase.

Another clear issue in the design of CS-Net and MRA-Net is that only data projected into the PAN domain are presented to the DCNNs. Namely, the inputs of the networks are practically monochromatic images i.e., without any spectral content). Thus, both the CS-Net and the MRA-Net receive no spectral information from these data. The networks fed in this way are not able to adequately reconstruct image features along the spectral direction, even training them with enough examples and a proper number of iterations. Instead, the use of $\text{PD} - \text{MS}$ as details to feed the network has the advantage to intrinsically introduce the spectral information. All these cues are supported by the experimental analysis showing that the Fusion-Net outperforms the other two proposed approaches.

Similar to the CS-Net and the MRA-Net, we ignore the injection coefficients $g$ in the general fusion equation of CS/MRA methods, allowing a DCNN to automatically estimate the nonlinear injection model. The Fusion-Net can be summarized as follows:

$$\text{MS} = \text{MS} + f_{\Theta_{\text{FS}}} (\text{PD} - \text{MS})$$

(8)

where $f_{\Theta_{\text{FS}}}$ is the nonlinear mapping with network parameters $\Theta_{\text{FS}}$.

Starting from (8), the network architecture of the proposed Fusion-Net is described in Fig. 4. In particular, the loss function for the Fusion-Net is as follows:

$$\text{Loss}(\Theta_{\text{FS}}) = \frac{1}{n} \sum_{k=1}^{n} \frac{\|\text{MS}^{(k)} + f_{\Theta_{\text{FS}}}(\text{PD}^{(k)} - \text{MS}^{(k)}) - \text{GT}^{(k)}\|_F^2}{\sum_{k=1}^{n} \|\text{MS}^{(k)}\|_F^2}$$

(9)

where the definitions of symbols are the same as that of (5).

Note that the Fusion-Net proposed in the work can be also regarded as a support strategy for deep learning (DP), thus improving the performance of existing methods. Please refer to Section IV for details about the performance gains.

D. Network Selection

We have proposed three deep network architectures for pansharpening, i.e., CS-Net, MRA-Net, and Fusion-Net. They all involved a subnetwork for training, i.e., “NetWork” (see the solid green boxes in Figs. 2–4). The main structure of “NetWork” is presented in Fig. 5(a). Wherein, we choose an effective network recently proposed in the literature, called ResNet [56], as the subnetwork of the proposed architectures since the ResNet can bring the conventional CNN to deeper layers leading to an effective and competitive performance in many image applications. Fig. 5 shows the basic structure of one ResNet block, in which one skip connection for every two convolutional layers is shown. In practical experiments, we need to empirically tune the number of ResNet blocks to control the final convolutional layers, aiming to achieve the best performance (see the parameter setting in Section IV).
In this work, we train the CNNs on WorldView-3 (eight bands) satellite data sets that can be easily downloaded on the public website. After downloading the data sets, we simulate 12,580 PAN/MS/GT image pairs with the size $64 \times 64$, $16 \times 16 \times 8$, and $64 \times 64 \times 8$, respectively, and then split them into 70/20/10% for training (8806 examples)/validation (2516 examples)/testing (1258 examples). Note that since the GT images are not available, we need to follow Wald’s protocol [57] to get them. The process of simulating the training data set by Wald’s protocol is illustrated in Fig. 6. It mainly contains the following steps: 1) downsampling the original PAN and the original MS image by a resolution factor 4 using modulation transfer function (MTF)-based filters, seeing the downsampled PAN image as the training PAN image and the downsampled MS image as the training MS image; 2) taking the original MS image as the training GT image; and 3) upsampling the training MS image by using a polynomial kernel with 23 coefficients [17] and interpreting the output as the upsampled MS image. Following steps 1–3, it is easy to generate the training data. The validation and testing data sets are similarly built.

IV. EXPERIMENTAL RESULTS

In this section, we compare the proposed network architectures (i.e., CS-Net, MRA-Net, and Fusion-Net) with some recent state-of-the-art pansharpening approaches belonging to the CS, the MRA, and the ML classes. First, the employed sensors, the benchmark, and the adopted quality indexes will be described. Afterward, the experimental analysis both at reduced and full resolutions will be described.

A. Data Sets

Several data sets have been acquired by the WorldView-2 and WorldView-3 sensors. The former provides a high-resolution PAN channel and eight MS bands. Four standard colors (red, green, blue, and near-infrared 1) and four new bands (coastal, yellow, red edge, and near-infrared 2) are acquired. Although the native spatial resolution would be greater, the images are distributed with a pixel size of 0.5 and 2 m for PAN and MS, respectively. The spatial ratio is equal to 4. The radiometric resolution is 11 bits.

WorldView-3 data have the same features as WorldView-2 data, but with a spatial resolution of about 0.3 m for the PAN channel and of about 1.2 m for the MS bands and a radiometric resolution of 11 bits. Moreover, we also assess the performance on four-band (red, green, blue, and near-infrared) data sets. In particular, QB data are considered having a spatial resolution of 2.4 and 0.61 m for the MS and PAN images, respectively, and a radiometric resolution of 11 bits.

Finally, images acquired by the GF-2 sensor have been exploited with a spatial resolution of 3.2 and 0.8 m for the MS and PAN images, respectively, and a radiometric resolution of ten bits (please see Section IV-H for more details).

B. Benchmark

The proposed benchmark consists of the following methods: the MS image interpolation using a polynomial kernel with 23 coefficients (EXP) [17], the GS sharpening approach [12], the SFIM [13], the PRACS approach [11], the BDSD method [9], the robust BDSD approach (BDSD-PC) [10], the GLP with MTF-matched filter [58] and multiplicative injection model [59] [GLP-high-pass modulation (HPM)], the GLP with MTF-matched filter [58] and regression-based injection model [GLP-context-based decision (CBD)] [3], [17], the GLP with full-scale regression (GLP-Reg) [20], the state-of-the-art CNN-based method for pansharpening (PNN) [41],

1http://www.digitalglobe.com/samples?search=Imagery

2We tried to simulate the same training data set as in [1] (PanNet), but, in the original article, the authors do not indicate which WorldView-3 data sets are selected for the training. However, in our work, all deep learning-based methods are trained on the same data set for a fair comparison.

3Note that the given source code in open remote sensing does not contain the trained models for WV2 and WV3; thus, we reimplemented the network with default parameters in Python using TensorFlow for simplicity of comparison.
TABLE I

<table>
<thead>
<tr>
<th>Param.</th>
<th>PNN</th>
<th>DRPNN</th>
<th>CS-Net</th>
<th>MRA-Net</th>
<th>DiCNN1</th>
<th>PanNet</th>
<th>DMDNet</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iter. #</td>
<td>$1.12 \times 10^6$</td>
<td>$3 \times 10^5$</td>
<td>$1.8 \times 10^5$</td>
<td>$1.6 \times 10^5$</td>
<td>$3 \times 10^5$</td>
<td>$2.4 \times 10^5$</td>
<td>$2.5 \times 10^5$</td>
<td>$1.4 \times 10^5$</td>
</tr>
<tr>
<td>BS</td>
<td>128</td>
<td>64</td>
<td>64</td>
<td>32</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Algo</td>
<td>SGD</td>
<td>SGD</td>
<td>Adam</td>
<td>Adam</td>
<td>Adam</td>
<td>Adam</td>
<td>Adam</td>
<td>Adam</td>
</tr>
<tr>
<td>LR</td>
<td>0.00001</td>
<td>0.05, 0.005</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0003</td>
</tr>
<tr>
<td>FS</td>
<td>$9 \times 9$, $5 \times 5$</td>
<td>$7 \times 7$</td>
<td>$3 \times 3$</td>
<td>$3 \times 3$</td>
<td>$3 \times 3$</td>
<td>$3 \times 3$</td>
<td>$3 \times 3$</td>
<td>$3 \times 3$</td>
</tr>
<tr>
<td>Filt. #</td>
<td>64, 32</td>
<td>64</td>
<td>32</td>
<td>64</td>
<td>32</td>
<td>64</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>N</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ly. #</td>
<td>3</td>
<td>11</td>
<td>10</td>
<td>18</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

We summarize the optimal parameters of all CNN methods in Table I.9

TABLE II

<table>
<thead>
<tr>
<th>Method</th>
<th>SAM (± std)</th>
<th>ERGAS (± std)</th>
<th>Q4 (± std)</th>
<th>SCC (± std)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNN</td>
<td>4.4015 ± 1.392</td>
<td>3.2923 ± 1.049</td>
<td>0.8693 ± 0.1123</td>
<td>0.9215 ± 0.0464</td>
</tr>
<tr>
<td>DRPNN</td>
<td>4.2657 ± 1.2531</td>
<td>3.0517 ± 0.9507</td>
<td>0.9010 ± 0.1098</td>
<td>0.9317 ± 0.0475</td>
</tr>
<tr>
<td>DiCNN1</td>
<td>3.9803 ± 1.3181</td>
<td>2.7567 ± 1.1556</td>
<td>0.9096 ± 0.1117</td>
<td>0.9517 ± 0.0471</td>
</tr>
<tr>
<td>PanNet</td>
<td>4.5931 ± 1.3723</td>
<td>2.8526 ± 0.9778</td>
<td>0.8541 ± 0.1170</td>
<td>0.9494 ± 0.0460</td>
</tr>
<tr>
<td>MRA-Net</td>
<td>3.9714 ± 1.3482</td>
<td>2.8572 ± 0.9663</td>
<td>0.9000 ± 0.1141</td>
<td>0.9527 ± 0.0446</td>
</tr>
<tr>
<td>DMDNet</td>
<td>4.4851 ± 1.4605</td>
<td>3.1036 ± 1.2124</td>
<td>0.8857 ± 0.1156</td>
<td>0.9388 ± 0.0509</td>
</tr>
<tr>
<td>Fusion-Net</td>
<td>4.5300 ± 1.4350</td>
<td>3.2857 ± 1.1169</td>
<td>0.8665 ± 0.1180</td>
<td>0.9372 ± 0.0482</td>
</tr>
<tr>
<td>Fusion-Net</td>
<td>3.7435 ± 1.2259</td>
<td>2.5679 ± 0.9441</td>
<td>0.9155 ± 0.1322</td>
<td>0.9580 ± 0.0450</td>
</tr>
</tbody>
</table>

We summarize the optimal parameters of all CNN methods in Table I.9

E. Reduced Resolution Assessment

After the training phase, we need to validate the performance of the compared CNN methods on WorldView-3 testing data. In this phase, we exclude classical CS and MRA methods because they will be strongly penalized by the absence of a training phase using similar samples that will be found in the testing data set. Thus, this analysis is only devoted to comparing CNN-based approaches trained on the same examples.

In Table II, we first show the average quantitative results of the different methods on the testing data set containing 1258 testing examples. For each testing example, the sizes of PAN, MS, and GT images are the same as that of the training examples, i.e., $64 \times 64$ for the PAN image, $16 \times 16 \times 8$ for the original low spatial resolution MS image, and $64 \times 64 \times 8$ for the GT image. From Table II, it is clear that the proposed Fusion-Net obtains the best average quantitative performance for all the quality indexes. Furthermore, the standard deviations (std) of all the metrics get the smallest values for all the indexes, which also demonstrates the robustness of the proposed Fusion-Net. In particular, having a look at Table II,

---

4It is not easy to find the source code; thus, we reimplemented the network with default parameters in Python using Tensorflow for simplicity of comparison.

5Code link: https://xueyangfu.github.io/

6DiCNN1 has been implemented by ourselves.

7DMDNet has been implemented by ourselves.

8http://openremotesensing.net/kb/codes/pansharpening/

9It is not easy to find the source code; thus, we reimplemented the network with default parameters in Python using Tensorflow for simplicity of comparison.
Fig. 7. Visual comparisons in natural colors of the most representative 13 approaches on the Rio data set (WorldView-3). (a) EXP. (b) GS. (c) SFIM. (d) BDSD. (e) BDSD-PC. (f) GLP-Reg. (g) GLP-CBD. (h) PNN. (i) DRPNN. (j) DiCNN1. (k) PanNet. (l) DMDNet. (m) Fusion-Net. (n) GT.

Fig. 8. AEMs of Fig. 7. (a) EXP. (b) GS. (c) SFIM. (d) BDSD. (e) BDSD-PC. (f) GLP-Reg. (g) GLP-CBD. (h) PNN. (i) DRPNN. (j) DiCNN1. (k) PanNet. (l) DMDNet. (m) Fusion-Net. (n) GT.

it is clear that the results of CS-Net and MRA-Net are worse than that of the recent DL-based methods (see the reasons underlined in the first two paragraphs in Section III-C). Hence, we will not show the results of CS-Net and MRA-Net from hereon, considering as a unique comparison the one with Fusion-Net. However, the presentation of CS-Net and MRA-Net is still meaningful since the proposed Fusion-Net is inspired and motivated by them.

A further test is about the use of two new WorldView-3 data sets capturing scenarios never presented to the networks in their training phase. In this case, the whole benchmark is used considering the comparison fair even when classical CS and MRA approaches are used. Again, Wald’s protocol is used to generate a reference (GT) image, as described in Section III-E. The two data sets will be named Rio and Tripoli from hereon, considering as 30-cm resolution. Their size is $256 \times 256 \times 8$ for the GT image, $256 \times 256$ for the PAN image, and $64 \times 64 \times 8$ for the original low spatial resolution MS image. Table III indicates that the best performance is still reached by the proposed Fusion-Net outperforming the performance of all the other compared pansharpening approaches for all the quality metrics. Similar conclusions can be drawn when the Tripoli data set is used.

The visual analysis further corroborates these numerical results. Indeed, in Fig. 7 (Rio data set), it is clear to see that the visual results provided by the classical CS and MRA methods (e.g., GS, SFIM, BDSD, BDSD-PC, GLP-Reg, and GLP-CBD) show low spatial performance with evident blur effects. Moreover, all six CNN methods perform significantly better than the classical methods (both spatially and spectrally). This demonstrates the ability of CNN methods to address the problem of pansharpening. It is worth to be remarked that it is not easy to distinguish the visual differences among the CNN methods in Fig. 7. This is due to the limitations in representing 8-bit RGB images instead of the 11-bit MS data. However, exploiting the calculation of the absolute error maps (AEMs) of Fig. 8, the visual advantages of the proposed Fusion-Net are pointed out getting lower
image residuals (see the close-up boxes in Fig. 8). The same conclusions can be drawn for the visual analysis of the fusion outcomes using the Tripoli data set in Figs. 9 and 10.

**F. Full-Resolution Assessment**

In this section, we test the performance of the proposed benchmark at the original (full) scale. In this case, the GT image is not available requiring quality indexes without reference for performance assessment purposes. We exploited 30 image pairs (MS and PAN) of WorldView-3 data at the original scale (OS) for testing the approaches using the QNR as the quality index. Table IV shows the quantitative assessment for all the methods in the benchmark. The six deep networks, i.e., PNN, DRPNN, DiCNN1, PanNet, DMDNet, and the proposed Fusion-Net, outperform the classical approaches. Having a look at the overall quality index QNR, the best average performance is obtained by the proposed Fusion-Net, even with a limited standard deviation implying that we got a robust result. The same can be stated for the spectral index $D_s$. Moreover, the best performance (comparable with the PanNet one) is obtained on the spatial index $D_s$. Finally, Fig. 11 shows the visual performance on a full-resolution WorldView-3 data set, here named the Tripoli-OS data set. It is easy to remark from Fig. 11 the lower spatial performance of the classical CS and MRA methods (e.g., GS, SFIM, BDSD, BDSD-PC, GLP-Reg, and GLP-CBD), whereas all the CNN-based methods significantly outperform the classical approaches, both spatially and spectrally. Furthermore, the proposed Fusion-Net obtains better spatial performance than that of the other five CNN-based methods. Meanwhile, Fusion-Net is also able to preserve the spectral information.

**G. Network Generalization**

We have demonstrated that the proposed Fusion-Net outperforms the other pansharpening approaches in the benchmark on WorldView-3 data when the networks are also trained on WorldView-3 data. In this section, we will focus on the capability of the networks to generalize the results
fusing data acquired by different sensors. To this aim, we exploit another data set acquired by another eight-band sensor, i.e., WorldView-2, but using the networks trained on WorldView-3 data. In order to have an accurate assessment, we still leverage on Wald’s protocol to generate the so-called Stockholm data set acquired by the WorldView2 sensor. Quantitative results reported in Table V indicate that the proposed Fusion-Net is again the best approach outperforming the benchmark on the metrics of ERGAS and Q8. The DMDNet obtains slightly better SAM and SCC metrics than Fusion-Net since it employs the dilated convolution that could significantly increase the receptive field, whereas our Fusion-Net only uses conventional convolution. Fig. 12 corroborates this statement. It is easy to see that all the CNN methods yield better spatial performance than the CS and MRA approaches. In Fig. 13, the AEMs of Fig. 12 are also shown. Again, the proposed Fusion-Net exhibits the darker residual map demonstrating its superiority with respect to the other compared approaches even from a qualitative point of view.

H. Assessment on Four-Band Data Sets

In this section, we will extend the performance assessment to four-band data sets, i.e., acquired by the GF-2 and the QB sensors.

About the data simulation, we also follow the way described in Section III-E to generate the training and testing data.
Fig. 12. Visual comparisons in natural colors of the most representative 13 approaches on the Stockholm data set (WorldView2). (a) EXP. (b) GS. (c) SFIM. (d) BDSD. (e) BDSD-PC. (f) GLP-Reg. (g) GLP-CBD. (h) PNN. (i) DRPNN. (j) DiCNN1. (k) PanNet. (l) DMDNet. (m) Fusion-Net. (n) GT.

Fig. 13. AEMs of Fig. 12. (a) EXP. (b) GS. (c) SFIM. (d) BDSD. (e) BDSD-PC. (f) GLP-Reg. (g) GLP-CBD. (h) PNN. (i) DRPNN. (j) DiCNN1. (k) PanNet. (l) DMDNet. (m) Fusion-Net. (n) GT.

TABLE V
QUANTITATIVE RESULTS ON THE STOCKHOLM DATA SET (WORLDVIEW2). BEST RESULTS ARE IN BOLDFACE

<table>
<thead>
<tr>
<th>Method</th>
<th>SAM</th>
<th>ERGAS</th>
<th>Q8</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP</td>
<td>7.589</td>
<td>9.079</td>
<td>0.6540</td>
<td>0.4508</td>
</tr>
<tr>
<td>GS</td>
<td>7.7296</td>
<td>7.3644</td>
<td>0.8075</td>
<td>0.8439</td>
</tr>
<tr>
<td>SFIM</td>
<td>7.1147</td>
<td>6.9570</td>
<td>0.8434</td>
<td>0.8562</td>
</tr>
<tr>
<td>BDSD</td>
<td>7.1824</td>
<td>6.3772</td>
<td>0.8798</td>
<td>0.860</td>
</tr>
<tr>
<td>BDSD-PC</td>
<td>7.0953</td>
<td>6.3233</td>
<td>0.8819</td>
<td>0.8578</td>
</tr>
<tr>
<td>PRACS</td>
<td>7.5894</td>
<td>7.4080</td>
<td>0.8314</td>
<td>0.8125</td>
</tr>
<tr>
<td>GLP-HPM</td>
<td>7.2988</td>
<td>6.9965</td>
<td>0.8527</td>
<td>0.8355</td>
</tr>
<tr>
<td>GLP-CBD</td>
<td>7.1098</td>
<td>6.5434</td>
<td>0.8752</td>
<td>0.8457</td>
</tr>
<tr>
<td>GLP-Reg</td>
<td>7.1195</td>
<td>6.4998</td>
<td>0.8776</td>
<td>0.8453</td>
</tr>
<tr>
<td>PNN</td>
<td>6.8604</td>
<td>5.6259</td>
<td>0.8642</td>
<td>0.8339</td>
</tr>
<tr>
<td>DRPNN</td>
<td>6.4798</td>
<td>5.6459</td>
<td>0.8843</td>
<td>0.8668</td>
</tr>
<tr>
<td>DiCNN1</td>
<td>6.8159</td>
<td>5.9773</td>
<td>0.8802</td>
<td>0.8797</td>
</tr>
<tr>
<td>PanNet</td>
<td>6.3916</td>
<td>5.6302</td>
<td>0.8897</td>
<td>0.8895</td>
</tr>
<tr>
<td>DMDNet</td>
<td>6.1986</td>
<td>5.5692</td>
<td>0.8903</td>
<td>0.8965</td>
</tr>
<tr>
<td>Fusion-Net</td>
<td>6.2784</td>
<td>5.5499</td>
<td>0.8969</td>
<td>0.8897</td>
</tr>
</tbody>
</table>

For the QB test case, we downloaded a large data set (4906 × 4906 × 4) acquired over the city of Indianapolis cutting it into two parts. The left part (4906 × 3906 × 4) is used to simulate 20,685 training samples (size: 64 × 64 × 4), and the right part (4906 × 1000 × 4) is used to simulate 48 testing data (size: 256 × 256 × 4). For the GF-2 test case, we downloaded a large data set (6907 × 7300 × 4) over the city of Beijing from the website to simulate 21,607 training examples (size: 64 × 64 × 4). Besides, a huge image acquired over the Guangzhou city is downloaded to simulate 81 testing data (size: 256 × 256 × 4).

Figs. 14 and 15 present the visual performance of the five representative CNN-based methods. The visual results provided by the six CNN methods all obtain competitive outcomes, both spatially and spectrally. As previously said, the RGB images shown in the first rows of Figs. 14 and 15 are not enough to show the differences of compared methods; thus, we calculate the AEMs in the second rows of Figs. 14 and 15.

---

10 Data link: http://www.rscloudmart.com/dataProduct/sample

11 Note that, since our CS-Net and MRA-Net get weak performance according to the results on WorldView-2 and WorldView-3 data sets, hence, for the sake of brevity, we excluded these two methods from the analysis. Furthermore, for the same reason, we only show the results of the five CNN methods.
to aid the visual comparison. From the two figures, the proposed Fusion-Net clearly shows its spatial advantages getting lower image residuals (see the close-up boxes). Moreover, from Table VI, the proposed Fusion-Net still yields better quantitative assessments than the other compared approaches.

I. Discussion

Based on the previously shown results, it is clear that the CNN methods obtain better performance than the classical CS and MRA methods. This is mainly due to the fact that these methods exploit large-scale data for the training phase. In this section, we will discuss more about the detail images, the convergence, the network complexity, the computational times in both testing and training phases, the number of parameters (NoPs), and the giga floating-point operations per second (GFLOPs).

1) Detail Images: Unlike the previously shown AEMs, Fig. 16 displays the detailed images in order to point out the differences among the compared methods. The detailed images are obtained by taking the absolute value of the difference between the fused and the EXP images. From Fig. 16, the Fusion-Net gets the darker detail image, which demonstrates the effectiveness of the proposed method even exploiting this different representation of the fused outcomes.

2) Convergence: Fig. 17 exhibits the training errors of all the deep network methods with increasing iterations. It is worth to be noted that the maximum number of iterations for each method is the corresponding optimal iteration. It is straightforward that the training error of the proposed Fusion-Net (black line) reaches a lower level than those of the other approaches, which demonstrates that the Fusion-Net gets better training effectiveness.

3) Network Complexity: The proposed Fusion-Net is simpler than the PanNet. Comparing it with PanNet, Fusion-Net does not need to calculate the high-pass filtered version of the PAN image, thus reducing the training time with respect
TABLE VI

<table>
<thead>
<tr>
<th></th>
<th>SAM (± std)</th>
<th>ERGAS (± std)</th>
<th>Q4 (± std)</th>
<th>SCC (± std)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guangzhou (GF-2)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNN</td>
<td>1.6599 ± 0.3606</td>
<td>1.5707 ± 0.3243</td>
<td>0.9274 ± 0.0202</td>
<td>0.9281 ± 0.0206</td>
</tr>
<tr>
<td>DRPNN</td>
<td>1.4578 ± 0.2289</td>
<td>1.3735 ± 0.1876</td>
<td>0.9308 ± 0.0148</td>
<td>0.9384 ± 0.0052</td>
</tr>
<tr>
<td>DiCNN1</td>
<td>1.4948 ± 0.3814</td>
<td>1.3203 ± 0.3543</td>
<td>0.9445 ± 0.0211</td>
<td>0.9458 ± 0.0222</td>
</tr>
<tr>
<td>PanNet</td>
<td>1.3954 ± 0.3261</td>
<td>1.2239 ± 0.2828</td>
<td>0.9463 ± 0.0222</td>
<td>0.9558 ± 0.0123</td>
</tr>
<tr>
<td>DMDNet</td>
<td>1.2968 ± 0.3156</td>
<td>1.1281 ± 0.2669</td>
<td>0.9529 ± 0.0218</td>
<td>0.9644 ± 0.0100</td>
</tr>
<tr>
<td>Fusion-Net</td>
<td>1.1795 ± 0.2714</td>
<td>1.0023 ± 0.2271</td>
<td>0.9627 ± 0.0167</td>
<td>0.9710 ± 0.0074</td>
</tr>
<tr>
<td><strong>Indianapolis dataset (QB)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNN</td>
<td>5.7993 ± 0.9474</td>
<td>5.3712 ± 0.4584</td>
<td>0.8572 ± 0.1481</td>
<td>0.9023 ± 0.0489</td>
</tr>
<tr>
<td>DRPNN</td>
<td>5.3667 ± 0.7721</td>
<td>5.270 ± 0.2809</td>
<td>0.8745 ± 0.1320</td>
<td>0.9177 ± 0.0454</td>
</tr>
<tr>
<td>DiCNN1</td>
<td>5.3071 ± 0.9957</td>
<td>5.231 ± 0.5411</td>
<td>0.8821 ± 0.1431</td>
<td>0.9224 ± 0.0586</td>
</tr>
<tr>
<td>PanNet</td>
<td>5.3144 ± 1.0175</td>
<td>5.1623 ± 0.6814</td>
<td>0.8833 ± 0.1398</td>
<td>0.9296 ± 0.0585</td>
</tr>
<tr>
<td>DMDNet</td>
<td>5.1197 ± 0.9399</td>
<td>4.7377 ± 0.6486</td>
<td>0.8907 ± 0.1464</td>
<td>0.9350 ± 0.0652</td>
</tr>
<tr>
<td>Fusion-Net</td>
<td>4.5402 ± 0.7789</td>
<td>4.0508 ± 0.2666</td>
<td>0.9102 ± 0.1364</td>
<td>0.9547 ± 0.0457</td>
</tr>
</tbody>
</table>

Fig. 16. Detail images of the different compared methods on a sample belonging to the Indianapolis data set (sensor: QB). (a) GS. (b) SFIM. (c) BDS. (d) BDS-PC. (e) GLP-Reg. (f) GLP-CBD. (g) PNN. (h) DRPNN. (i) DiCNN1. (j) PanNet. (k) DMDNet. (l) Fusion-Net.

Fig. 17. Convergence curves for all the compared CNN methods on the WorldView-3 training data set. Note that we trained the PNN method with $1.12 \times 10^6$ iterations, but, here, we only show MSEs of the first $3 \times 10^5$ iterations for better display.

to PanNet. The architecture of DMDNet is similar to that of the PanNet, but DMDNet has a structure of grouped dilated convolution and, thus, is more complicated than PanNet. The architecture of DiCNN1 is slightly simpler than the PanNet and the Fusion-Net, but it is only a three-layer network meaning that it is not easy to extract sufficient image features. The architecture of PNN is a simple three-layer network without any skip connection; thus, it is also not easy to extract enough image features from the simple network. In addition, the architecture of the DRPNN contains a skip connection and 11 layers, thus having a better feature extraction ability.

4) Testing Time: Table III reports the testing time of all the compared methods on two WorldView-3 data (i.e., Rio data set and Tripoli data set, both with size $256 \times 256 \times 8$). Classical CS and MRA methods generally reach shorter testing times than that of the CNN methods. Furthermore, it is worth to be noted that CNN times are calculated on special hardware architecture (GPU); instead, to calculate the times of the CS and MRA approaches, a general-purpose CPU has been used. However, the testing time of the proposed networks can be considered acceptable on these data.

5) Training Time: The training times of all the CNNs are reported using the same training data set. The maximum
Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 20,2021 at 11:17:02 UTC from IEEE Xplore. Restrictions apply.


Liang-Jian Deng (Member, IEEE) received the B.S. and Ph.D. degrees in applied mathematics from the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2016, respectively.

From 2013 to 2014, he was a Joint-Training Ph.D. Student with Case Western Reserve University, Cleveland, OH, USA. From 2017 to 2018, he was a Post-Doctorate with Hong Kong Baptist University (HKBU), Hong Kong. In addition, he also stayed with the Isaac Newton Institute for Mathematical Sciences (Cambridge University), Cambridge, U.K., and HKBU for short visits. He is an Associate Professor with the School of Mathematical Sciences, UESTC. His research interests include the use of partial differential equations (PDE), optimization modeling, and deep learning to address several tasks in image processing and computer vision, e.g., resolution enhancement, restoration, etc.

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, Salerno, Italy, in 2008, 2011, and 2014, respectively.

In 2013, he was a Visiting Scholar with the Grenoble Institute of Technology (INPG), Grenoble, France. 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. In 2019, he was an Assistant Professor with the University of Salerno. He is the Leader of the Image and Signal Processing Working Group of the IEEE Image Analysis and Data Fusion Technical Committee. He is also a Researcher with the National Research Council, Rome, Italy, and an Adjunct Professor with the University of Salerno. His main research interests focus on statistical signal processing, detection of remotely sensed images, data fusion, and tracking algorithms.

Dr. Vivone is an Editorial Board Member for Multidisciplinary Digital Publishing Institute (MDPI) Remote Sensing, MDPI Sensors, and MDPI Encyclopedia. He received 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 (GRS) in 2017. He served as a Guest Associate Editor for several special issues. He is an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL).

Cheng Jin is pursuing the B.S. degree with the School of Optoelectronic Science and Technology, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include digital image processing utilizing deep learning, e.g., resolution enhancement, etc.

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. He was a Visiting Scholar with Stanford University, Stanford, CA, USA, KTH, Stockholm, Sweden, and National University of Singapore (NUS), Singapore. Since 2013, he has 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 with the Chinese Academy of Sciences, Aerospace Information Research Institute, Beijing, China. He is a Professor of signal and image processing with Grenoble INP. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning, and artificial intelligence.

Dr. Chanussot was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society from 2006 to 2008, and the Institut Universitaire de France from 2012 to 2017. He is the Founding President of the IEEE Geoscience and Remote Sensing (GRS) French chapter from 2007 to 2010 which received the 2010 IEEE Geoscience and Remote Sensing (GRS)-S Chapter Excellence Award. He has received multiple outstanding paper awards. He was the Vice-President of the IEEE GRS Society (GRSS), the In-Charge of meetings and symposia from 2017 to 2019. He was the General Chair of 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 of the GRS Data Fusion Technical Committee from 2005 to 2008. He was the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2009. He is an Associate Editor for 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 is a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters from 2018 to 2019.