Machine Learning in Pansharpening: A Benchmark, From Shallow to Deep Networks

Article *in* IEEE Geoscience and Remote Sensing Magazine · September 2022

DOI: 10.1109/MGRS.2022.3187652

Liang-Jian Deng
University of Electronic Science and Technology of China
93 PUBLICATIONS **2,505** CITATIONS

Gemine Vivone
Università degli Studi di Salerno
161 PUBLICATIONS **5,913** CITATIONS

Mercedes Paoletti
Universidad de Extremadura
96 PUBLICATIONS **4,259** CITATIONS

Jiang He
Wuhan University
27 PUBLICATIONS **465** CITATIONS

Machine Learning in Pansharpening: A Benchmark, from Shallow to Deep Networks

Liang-Jian Deng, *Member, IEEE,* Gemine Vivone, *Senior Member, IEEE,*
Mercedes E. Paoletti, *Senior Member, IEEE,* Giuseppe Scarpa, *Senior Member, IEEE,* Jiang He,
Yongjun Zhang, *Member, IEEE,* Jocelyn Chanussot, *Fellow, IEEE,* Antonio Plaza, *Fellow, IEEE*

***Abstract*—Machine learning is influencing the literature in several research fields, often proposing state-of-the-art approaches. In the last years, machine learning has been explored even for pansharpening, *i.e.,* an image fusion technique based on the combination of a multispectral image, which is characterized by its medium/low spatial resolution, and a higher spatial resolution panchromatic data. Thus, machine learning for pansharpening represents an emerging research line that deserves further investigations. In this work, we go through some powerful and widely used machine learning-based approaches for pansharpening recently proposed in the related literature. Eight approaches have been extensively compared. Implementations of these eight methods exploiting a common software platform and machine learning library are developed for comparison purposes. The machine learning framework for pansharpening will be freely distributed to the scientific community. Experimental results using data acquired by five commonly used sensors for pansharpening and well-established protocols for performance assessment (both at reduced resolution and at full resolution) are shown. The machine learning-based approaches are compared with a benchmark consisting of classical and variational optimization-based methods. The pros and the cons of each pansharpening technique based on the training by examples philosophy are reported together with a broad computational analysis.**

***Index Terms*—Benchmarking, Convolutional Neural Networks, Deep Learning, Machine Learning, Pansharpening, Quality Assessment, Very High-resolution Optical Images, Image Fusion, Remote Sensing.**

L.-J. Deng is with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China (e-mail: liangjian.deng@uestc.edu.cn).

G. Vivone is with the Institute of Methodologies for Environmental Analysis, CNR-IMAA, Tito Scalo, 85050, Italy (e-mail: gemine.vivone@imaa.cnr.it).

M. E. Paoletti is with the Department of Computer Architecture and Automatics of the Faculty of Computer Science, Complutense University of Madrid, Madrid, 28040, Spain (e-mail: mpaolett@ucm.es).

G. Scarpa is with the Department of Electrical Engineering and Information Technology, University Federico II, Naples, 80125, Italy (e-mail: giscarpa@unina.it).

J. He is with the School of Geodesy and Geomatics, Wuhan University, Hubei, 430079, China (e-mail: jiang_he@whu.edu.cn).

Y. Zhang is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, 430079, China (e-mail: zhangyj@whu.edu.cn).

J. Chanussot is with Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble, 38000, France (e-mail: jocelyn.chanussot@gipsa-lab.grenoble-inp.fr).

A. Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, 10003, Spain (e-mail: aplaza@unex.es).

I. Introduction

Pansharpening is the process of combining a multispectral (MS) image with a panchromatic (PAN) image to produce an output that holds the same spatial resolution as the PAN image and the same spectral resolution as the MS image. To date, several techniques dealing with this problem have been proposed. With the development of new hardware and software solutions, machine learning (ML) approaches, especially deep learning-based (DL) frameworks, have been significantly developed. However, a fair comparison among these techniques (including, for instance, their development under the same software platform using same libraries, testing on datasets simulated in a conventional way, and so forth) is still an open issue. To this aim, in this article, we will go through from shallow to deep networks based on widely used and powerful ML-based pansharpening approaches. Besides, traditional approaches, belonging to component substitution (CS), multiresolution analysis (MRA), and variational optimization-based (VO), will also be compared and discussed. A quantitative and qualitative assessment will be presented in the experimental section, exploiting protocols both at reduced resolution and at full resolution. All the compared ML-based techniques have been implemented using Pytorch. The source codes will be freely distributed to the Community[1]. The ML framework for pansharpening uses an uniform programming style to facilitate the interpretability for users.

A. Background & Related Works

Recently, some books [1] and review articles [2]–[4] about pansharpening have been published attesting to the key role in the field of remote sensing image fusion. Besides, in recent years, some other surveys have been published, such as [5]–[7], confirming the increased interest in this area.

Many techniques have been applied to the task of remote sensing pansharpening. They are usually divided into four classes [4], *i.e.,* CS, MRA, VO, and ML. In the paper, we consider the first three classes as traditional methods since the first approaches have been proposed long time ago. In the meanwhile, several works in the related literature, such as, [2], [4], have deeply analyzed these categories. The remaining methods, belonging to the ML class, will be further investigated in this article. In the rest of this section, we will go through the four main categories of pansharpening algorithms introducing the related literature.

[1] Project page: *The link will be added after the paper acceptance.*

1) CS: CS approaches (also called spectral methods) rely on the projection into a transformed domain of the original MS image to separate its spatial information and substituting it with the PAN image. Many pioneering pansharpening techniques belong to the CS class thanks to their easy implementations. Two examples of CS approaches proposed in the early 1990s are the intensity-hue-saturation (IHS) [8], [9] and the principal component analysis (PCA) [10], [11].

By considering the various image transformations, a variety of techniques for incorporating PAN spatial information into the original MS data have been developed. These methods are usually viewed as the second generation of CS techniques, mainly improving the injection rules by investigating the relationship between the pixel values of the PAN image and those of the MS channels. Representative approaches are the Gram-Schmidt (GS) [12] and its adaptive version [13], the nonlinear PCA [14], and the partial replacement adaptive CS (PRACS) method [15].

Beyond the above-mentioned CS strategies, some other recent approaches are based on i) the local application of CS algorithms and ii) the joint estimation of both detail injection and estimation coefficients. The former subclass is mainly about sliding widow-based methods [13] or approaches relied upon clustering and segmentation [16], whereas the latter one includes band-dependent spatial-detail (BDSD) methods (see, *e.g.,* the BDSD [17] and its robust version [18]).

2) MRA: MRA methods apply a multiscale decomposition to the PAN image to extract its spatial components. This class is also referred to as spatial methods as they work into the spatial domain. General-purpose decompositions have been considered in the pansharpening literature, including, for instance, Laplacian pyramids [19], wavelets [20], curvelets [21], and contourlets [22]. MRA-based fusion techniques present interesting features, such as, temporal coherence [23], spectral consistency [2], and robustness to aliasing [24], thus deserving further investigations in subsequent years.

Recently, researchers have considered various decomposition schemes and several ways to optimize the injection model to improve MRA-based methods. Due to its superior performance in other image processing fields, nonlinear methods have also been introduced into pansharpening; typical examples are least-squares support vector machines [25] and morphological filters (MF) [26]. Moreover, thanks to an in-depth analysis of the relationship between the obtained images [27], [28] and the influence of the atmosphere on the collected signals, a series of advanced injection models has been designed [27], [29], [30]. A further crucial step forward has been the introduction of information about the acquisition sensors, thus driving the decomposition phase [24], [31]. This symbolized the beginning of the second generation for MRA-based pansharpening. The application of adaptive techniques has been proposed to deal with unknown or hardly predictable features about acquisition sensors [32], [33] and to address the peculiarities of some target images [34].

Hybrid technologies, combining MRA and CS methods, see *e.g.,* [4], have also been proposed. They can be regarded as MRA methods [24]. Within this category, two attempts have been considered, *i.e.,* "MRA+CS" (MRA followed by CS) [35] and "CS+MRA" (CS followed by MRA) [27], [36]. Other notable examples in this subclass include the use of independent component analysis in combination with curvelets [37] and the use of PCA with contourlets [38] or guided filters [39].

3) VO: The class of VO methods focuses on the solution of optimization models. In recent years, they have become more and more popular thanks to the advances in convex optimization and inverse problems, such as multispectral pansharpening [40]–[44] and hyperspectral image fusion [45]–[47]. Most of VO methods focus on the relationship between the input PAN image, the low spatial resolution MS (LRMS) image, and the desired high spatial resolution MS (HRMS) image to generate the corresponding model. However, the problem to be solved is clearly ill-posed, thus requiring some regularizers introducing prior information about the solution (*i.e.,* the HRMS). The target image is usually estimated under the assumption of proper co-registered PAN and LRMS images. Anyway, some papers, see, *e.g.,* [48], have been proposed to deal with registration issues.

The timeline of VO techniques starts from 2006, with the so-called P+XS method [49]. Inspired by P+XS, researchers have proposed various regularization terms [50], [51] and new fidelity terms [52]–[54]. In [55], authors indirectly model the relationship between PAN and HRMS images by considering the spectral low-rank relationship between them.

Apart from P+XS-like methods, other kinds of approaches belonging to the VO class mainly include Bayesian methods [56]–[59] and sparse representations [60]–[67].

4) ML: ML-based methods have shown a great ability in fusing MS and PAN data thanks to the recent advances in computer hardware and algorithms. Classical ML approaches mainly include dictionary learning methods [62]–[65] and compressive sensing techniques [60], [61]. Compressed sensing is about acquiring and reconstructing a signal by efficiently solving underdetermined linear systems. The sparsity of a signal can be utilized to recover the signal through proper optimization, even with considerably fewer samples than the ones required by the Nyquist-Shannon sampling theorem. The main stream based on compressive sensing pansharpening views the linear observation models (both the one about the LRMS and the one related to the PAN) as a measurement process in compressive sensing theory, then building effective and efficient algorithms to solve the related models under the sparsity assumption. Dictionary learning, a special representation strategy, is mainly based on sparse coding to find a sparse linear representation from the input data, forming a so-called dictionary matrix and the corresponding coefficients. The main idea of dictionary learning for pansharpening is to calculate (trained or not trained) dictionaries of LRMS and PAN images, then reconstructing the final HRMS pansharpened image by investigating the relation between dictionaries and the corresponding coefficients.

Recently, DL techniques have swept over almost all the applications in remote sensing image, even including multispectral pansharpening [68]–[83] and some closely related tasks like remote sensing image super-resolution [84]–[86] or hyperspectral image fusion [87]–[89]. The first work using a DL technique for pansharpening dates back to 2015 by Huang *et al.* [68], in which the autoencoder scheme inspired by the sparse denoising task was employed and modified. In 2016, Masi *et al.* [69] built and trained the first fully convolutional neural network (CNN) for pansharpening, also called pansharpening neural network (PNN). The architecture mainly consists of three convolutional layers, which is inspired by the super-resolution CNN [90] whose task was about the single image super-resolution problem. In the meanwhile, Zhong *et al.* [70] in 2016 also proposed a new CS pansharpening method based on the GS transform, in which a commercially available super-resolution CNN was exploited to upsample the MS component. After these pioneering approaches, this topic has received the interest of many researchers, as testified by a lots of publications, such as, [72], [76], [77], [81]–[83]. Thus, the use of CNNs has become a common choice for DL-based pansharpening. Unlike the PNN that has a simple network architecture, the subsequent pansharpening architectures have been deepened and widen, getting more and more complex structures with many parameters to fit during the training phase to obtain superior performance. These methods can be found in [71], [75], [79]. Besides, another research line using residual learning has been developed to effectively alleviate the phenomenon of gradient vanishing and explosion, thus accelerating the network convergence. Hence, the residual learning has been widely applied to pansharpening, see *e.g.,* [73], [91], [92]. A weak generalization ability of ML-based approaches can easily be observed representing a key issue. Therefore, another research line is going towards the development and the designing of new network architectures or pre-processing operators aiming to improve the generalization ability, see *e.g.,* [73], [74].

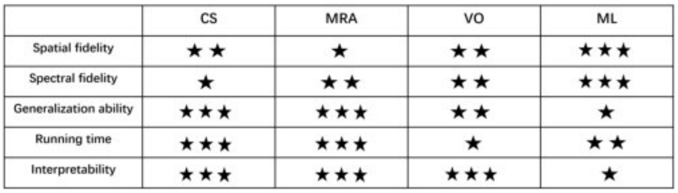
Except for the above-mentioned DL methods, some hybrid methods to combine traditional techniques (*e.g.*, CS, MRA and VO methods) and ML methods have recently become a promising direction in the field of remote sensing pansharpening, see *e.g.*, [47], [92]–[100]. For example, in [92], motivated by avoiding linear injection models and replacing the details injection phases in both CS and MRA methods, Deng *et al.* design a deep convolutional neural network, inspired by the CS and MRA schemes, to effectively manage the nonlinear mapping and extract image features, thus yielding favorable performance. In addition, with the development of DL and VO techniques, the literature is also presenting combinations of these two classes. Three strategies have been developed: the unfolding VO model [97], the plug-and-play operator [93], and the VO+Net mixed model [47], [96], which can also be viewed as belonging to the VO class. The outcomes of these latter approaches can benefit of both the advantages of DL and VO classes, *e.g.,* the good generalization ability of VO methods and the high performance of DL approaches. Specifically, in [94], Shen *et al.* incorporate the pansharpened outcomes learned from the DL model into a VO framework. This strategy is simple but quite effective in practical applications. Besides, Xie *et al.* in [95] also take the similar strategy as [94] for the task of hyperspectral pansharpening, still producing promising outcomes. Different from the strategy in [94], [95], new DL network architectures propose to unfold traditional VO models. In [97], Feng *et al.* present first a two-step optimization model based on spatial detail decomposition, then unfolding the given model under the gradient descent framework to further construct the corresponding end-to-end CNN architecture. Similarly to [97], Xu *et al.* in [98] propose a model-driven deep pansharpening network by gradient projection. Specifically, two optimization problems regularized by the deep prior are formulated. The two problems are solved by a gradient projection algorithm, in which the iterative steps are constructed by two network blocks that will be effectively trained in an end-to-end manner. Moreover, Cao *et al.* in [99] and Yin *et al.* in [100] present sparse coding based strategies to unfold the optimization models into subproblems which are replaced by learnable networks.

Recently, unsupervised learning strategy is introduced to the field of pansharpening, see *e.g.*, [101]–[103]. Unsupervised learning explores hidden patterns and features without any labeled data, which means that there is no need to simulate datasets with labels for training. It is a direct way for the network training but strongly dependent on the effectiveness of the loss function. In [101], Ma *et al.* propose a novel unsupervised pansharpening approach that can avoid the degrading effect of downsampling high-resolution MS images. Also, it considers the GAN strategy getting excellent results, in particular, on full-resolution data. Furthermore, Qu *et al.* in [103] present a self-attention mechanism-based unsupervised learning technique for pansharpening. This method can address some challenges, *e.g.*, poor performance on full-resolution images and wide presence of mixed pixels. In [104], leveraging on the target-adaptive strategy introduced in [74], Ciotola *et al.* present an unsupervised full-resolution training framework, demonstrating its effectiveness on different CNN architectures [71], [73], [74].

Moreover, the generative adversarial network (GAN) techniques [105] have recently been applied to the field of image processing. GAN is mainly about learning generative models via an adversarial process; thus, two models are required to be trained simultaneously, *i.e.*, generative models to capture data distribution and adversarial models to compute the probability of a sample to belong to training data or not. Especially, GANs have also been applied to the task of pansharpening, see *e.g.*, [78], [101], [106]–[110]. In [78], Liu *et al.* utilize first a GAN to address the task of remote sensing pansharpening, called PSGAN. This method mainly contains a two-stream fusion architecture consisting of a generator to produce the desired HRMS image and a discriminator to judge if the image is real or pansharpened. Instead, in [110], to further boost the accuracy, the authors propose a GAN-based pansharpening framework containing two discriminators, the first one dealing with image textures and the second one accounting for image color.

Finally, Tab. I gives an overview about the four classes described above focusing on some aspects, such as, the spatial fidelity, the spectral fidelity, the generalization ability, the running time, and the model interpretability. Just for example, it is easy to remark that ML methods generally get the best spatial and spectral performance, but requiring that training and testing data have similar properties (*e.g.,* a similar geo-

TABLE I: An overview about the pros and the cons of the four pansharpening classes. Weak: ★; Moderate: ★★; Strong: ★★★.



	CS	MRA	VO	ML
Spatial fidelity	★★	★	★★	★★★
Spectral fidelity	★	★★	★★	★★★
Generalization ability	★★★	★★★	★★	★
Running time	★★★	★★★	★	★★
Interpretability	★★★	★★★	★★★	★

graphic area and acquisition time).

B. Contribution

This paper is focused on a deep analysis of the emerging class of pansharpening algorithms based on ML paradigms. A complete review of the related literature has been presented. Afterwards, the paper will rely upon the critical comparison among state-of-the-art approaches belonging to the ML class. To this aim, a toolbox exploiting a common software platform and open-source ML library for all the ML approaches has been developed. We would like to stress that this is the only way to get a critical comparison of ML approaches. In fact, by changing software platforms and/or ML libraries (*e.g.,* Tensorflow or Caffe), we have different build-in functions, thus getting different behaviors (*e.g.,* a different initialization of the weights of the network) of the same algorithm coded in a different environment.

To overcome this limitation, a Python toolbox based on the Pytorch ML library (widely used for applications such as computer vision and natural language processing) has been developed. The toolbox will be freely distributed to the scientific community related to ML and pansharpening. Eight state-of-the-art approaches have been selected and implemented in the common framework following the original implementations proposed in the related papers. A tuning phase to ensure the highest performance for each approach has been performed. This latter represents a mandatory step to have a fair comparison because the eight approaches have been originally developed on different software platforms and/or using different ML libraries. A broad experimental analysis, exploiting different test cases, has been conducted with the aim of assessing the performance of each ML-based state-of-the-art approach. Widely used sensors for pansharpening have been involved (*i.e.,* WorldView-2, WorldView-3, WorldView-4, QuickBird, and IKONOS). The assessments both at reduced resolution and at full resolution have been exploited. Two test cases at reduced resolution have been considered. The first test is about the use of a part of the training set not used to this aim. However, by taking into account a testing area very close to the ones used in the training phase, we have a sort of coupling among data (*e.g.,* sharing features with the training samples like the atmospheric composition and conditions). Thus, to test the ability of the networks to work in a real scenario, we consider a second test case where the images are acquired by the same sensor but on a different area and at a different time with respect to the data used for the training. The comparison among ML-based approaches has also been enlarged to state-of-the-art methods belonging to different paradigms (*i.e.,* CS, MRA, and VO) exploiting standard implementations [4]. Finally, a wide computational analysis is presented to the readers. Execution times for training and testing, convergence analysis, number of parameters, and so forth have been highlighted. Moreover, the generalization ability of the networks with respect to the change of the acquisition sensor has also been discussed.

C. Notation & Organization

The notation is as follows. Vectors are indicated in bold lowercase (*e.g.,* $\mathbf{x}$) with the i-th element indicated as x_i. Two- and three-dimensional arrays are expressed in bold uppercase (*e.g.,* $\mathbf{X}$). An MS image $\mathbf{X} = \{\mathbf{X}_k\}_{k=1,\ldots,N}$ is a three dimensional array composed by N bands indexed by the subscript $k = 1, \ldots, N$; accordingly, $\mathbf{X}_k$ indicates the k-th band of $\mathbf{X}$. The PAN image is a 2-D matrix and will be indicated as $\mathbf{P}$. $\mathbf{MS}$ is the MS image, $\widetilde{\mathbf{MS}}$ is the MS image upsampled to the PAN scale, and $\widehat{\mathbf{MS}}$ is the fused image. The other symbols will be defined within the paper upon need.

The rest of the paper is organized as follows. Sect. II shows a brief overview of CS, MRA, and VO approaches detailing the ones exploited in this paper. Sect. III is about the ML-based methods belonging to the developed toolbox and compared in this work. Finally, experimental results showing performance on several datasets acquired by some of the most used sensors for pansharpening are reported in Sect. IV together with a computational analysis and some final remarks. Concluding remarks are instead drawn in Sect. V.

II. CS, MRA, and VO: A Brief Overview

In this section, we will go through the component substitution (CS), the multi-resolution analysis (MRA), and the variational optimization-based (VO) categories providing a brief overview for each class together with some instances of methods that have been exploited in this paper for comparison purposes.

The methods belonging to the CS class rely upon the projection of the MS image into a new space, where the spatial structure is separated from the spectral information [111]. Afterwards, the transformed MS image can be sharpened by substituting the spatial component with the PAN image. Finally, the sharpening process is completed by the inverse transformation to come back to the original space. CS methods get high fidelity in rendering details. Moreover, they are usually easy to implement and with a limited computational burden [2], [4].

Under the hypotheses of linear transformation and the substitution of a unique component, the CS fusion process can be simplified obtaining a faster implementation described by the following formulation [112]

$$\widehat{\mathbf{MS}}_k = \widetilde{\mathbf{MS}}_k + \mathbf{G}_k \cdot (\mathbf{P} - \mathbf{I}_L), \qquad (1)$$

in which $\widehat{\mathbf{MS}}_k$ is the k-th fused band, $\widetilde{\mathbf{MS}}_k$ is the upsampled image to the PAN scale, $\mathbf{P}$ is the PAN image, $\mathbf{G}_k$ is the injection gain matrix, the matrix multiplication is meant pointwise,

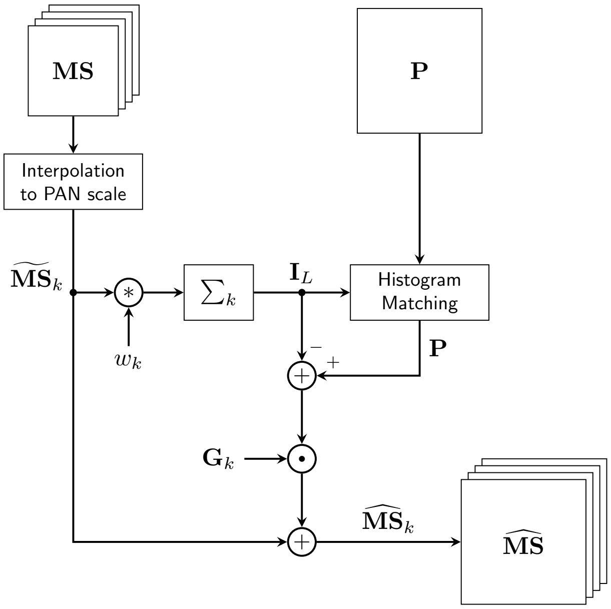


Fig. 1: Flowchart of CS-based methods.

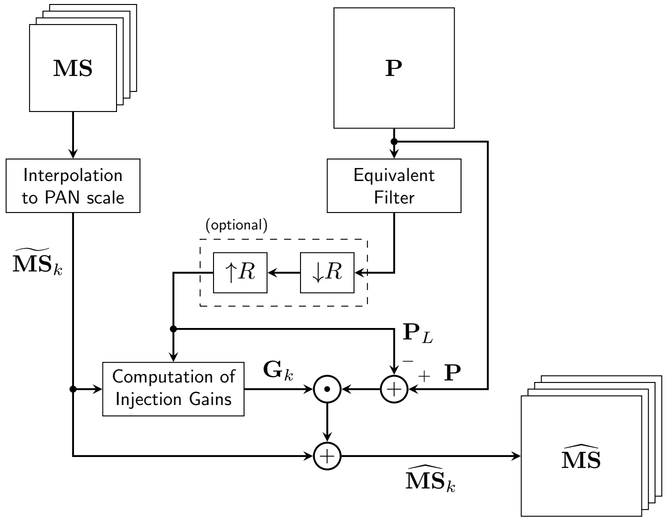


Fig. 2: Flowchart of MRA-based methods. Some MRA approaches skip the dashed box.

and $\mathbf{I}_L$ is the so-called intensity component obtained by a weighted average of the MS spectral bands with weights w_k.

Fig. 1 shows a flowchart describing the general fusion process for CS-based approaches. We can note blocks about the upsampling, the computation of $\mathbf{I}_L$, the spectral matching between $\mathbf{P}$ and $\mathbf{I}_L$, and the detail injection according to (1). Setting the injection gains in (1) as the pixel-wise division between $\widetilde{\mathbf{MS}}_k$ and $\mathbf{I}_L$, we have a multiplicative injection scheme (the widely known Brovey transform [113]) [114]. An interpretation of the Brovey transform in terms of the radiative transfer model led to the development of a haze-corrected version, called BT-H, recently proposed in [30]. The Gram-Schmidt (GS) orthogonalization procedure has also been used for pansharpening [115]. This procedure exploits the intensity component, $\mathbf{I}_L$, as the first vector of the new orthogonal basis. Pansharpening is obtained thanks to the substitution of $\mathbf{I}_L$ with the PAN image before inverting the transformation. Several versions of GS are achieved by varying $\mathbf{I}_L$. The context-adaptive GSA (C-GSA) is obtained by applying an adaptive GS (where the $\mathbf{I}_L$ is got by a weighted average of the MS bands using proper weights [116]) separately to each cluster [16]. The band dependent spatial detail (BDSD) framework, proposed for pansharpening in [17], exploits an extended version of (1) optimizing the minimum mean squared error (MMSE) for jointly estimating the weights and the scalar gains [17]. A physically constrained optimization (*i.e.,* the BDSD-PC) has recently been proposed in [117].

MRA methods extract the PAN details exploiting the difference between $\mathbf{P}$ and its low-pass spatial version, $\mathbf{P}_L$. The fused image is obtained as follows

$$\widehat{\mathbf{MS}}_k = \widetilde{\mathbf{MS}}_k + \mathbf{G}_k \cdot (\mathbf{P} - \mathbf{P}_L) . \tag{2}$$

These different approaches are characterized by the way in which they calculate $\mathbf{P}_L$ and to estimate the injection gains $\mathbf{G}_k$. In a very general setting, $\mathbf{P}_L$ is achieved through an iterative decomposition scheme, called MRA.

The general fusion scheme is depicted in Fig. 2. We can remark blocks devoted to the upsampling, the calculation of the low-pass version $\mathbf{P}_L$ of the PAN image based on the resolution ratio R, and the computation of the injection gains $\mathbf{G}_k$. MRA algorithms reduce the spectral distortion but often paying it with a greater spatial distortion [2], [4]. Among all the MRA approaches, one subcategory very debated is the one based on the generalized Laplacian pyramid (GLP). In this case, $\mathbf{P}_L$ can be performed with multiple fractional steps utilizing Gaussian low-pass filters to carry out the analysis steps [19]. The corresponding differential representation is called Laplacian pyramid. However, high performance can be obtained with a single Gaussian low-pass filter (tuned to closely match the MS sensor's modulation transfer function (MTF) [31]) with a cut frequency equal to $1/R$ (where R is the resolution ratio between PAN and MS) and decimating by R [4]. In the literature, many instances of GLP approaches, relying upon filters that exploit the MS sensor's MTF, have been proposed by changing the way to estimate the injection coefficients. We will exploit in this paper the high-pass modulation (HPM) injection [114], *i.e.,* setting the injection gains as the pixel-wise division between $\widetilde{\mathbf{MS}}_k$ and $\mathbf{P}_L$, adopting a spectral matching procedure based on the multivariate linear regression between each MS band and a low-pass version of the PAN image, *i.e.,* the MTF-GLP-HPM-R [118]. Moreover, we will also consider the MTF-GLP-FS that is based on an at full scale (FS) fusion rule, thus removing the hypothesis of invariance among scales for the coefficient injection estimation phase [119].

The methods in the VO category relied upon the definition of an optimization model. We will exploit two instances of techniques belonging to the concepts of sparse representation and total variation. In [66], an example of sparse representation for pansharpening is provided. In particular, the authors propose to generate the spatial details by using a dictionary of patches. Specifically, the dictionary $\mathbf{D}^h$ at full scale is composed of patches representing high spatial resolution details.

The coefficients $\boldsymbol{\alpha}$ of the linear combination are estimated by solving a sparse regression problem. Under the hypothesis of scale invariance, the coefficients can be estimated thanks to a reference image. The problem to solve is as follows

$$\hat{\boldsymbol{\alpha}} = \text{arg min } ||\boldsymbol{\alpha}||_0 \quad \text{such that} \quad \mathbf{y} = \mathbf{D}^l \boldsymbol{\alpha}, \tag{3}$$

where $\mathbf{y}$ is a patch, $||\cdot||_0$ is the l_0 norm, and $\mathbf{D}^l$ is a dictionary of details at reduced resolution. The estimated coefficients will be used for the representation of the full resolution details (*i.e.,* $\mathbf{y}^h = \mathbf{D}^h \boldsymbol{\alpha}$).

The cost function for the total variation (TV) pansharpening method in [50] is given by the following TV-regularized least squares problem

$$J(\mathbf{x}) = \|\mathbf{y} - \mathbf{M}\mathbf{x}\|^2 + \lambda \text{TV}(\mathbf{x}), \tag{4}$$

where $\mathbf{y} = [\mathbf{y}_{MS}^T, \mathbf{y}_{PAN}^T]$, $\mathbf{y}_{MS}$ and $\mathbf{y}_{PAN}$ are the MS in matrix format and the PAN in vector, $\mathbf{M} = [\mathbf{M}_1^T, \mathbf{M}_2^T]$, $\mathbf{M}_1$ is a decimation matrix, $\mathbf{M}_2$ reflecting that the PAN image is assumed to be a linear combination of the MS bands, λ is a weight, and $\text{TV}(\cdot)$ is an isotropic TV regularizer. The pansharpened image $\mathbf{x}$ is obtained by minimizing the convex cost function in (4).

III. A Benchmark Relied Upon Recent Advances in ML for Pansharpening

ML for pansharpening is mainly about the DL philosophy, as already pointed out in Sect. I-A. The approaches in this class strongly depend on the reduced resolution training set (or the full resolution one if belonging to the unsupervised paradigm). The testing datasets are exploited to get the network outcomes by using the trained models. In what follows, we chose eight representative supervised ML pansharpening approaches, *i.e.,* deep CNN architecture for pansharpening (PanNet) [73], deep residual neural network for pansharpening (DRPNN) [71], multiscale and multidepth CNN architecture for pansharpening (MSDCNN) [75], bidirectional pansharpening network (BDPN) [79], detail injection based convolutional neural network (DiCNN) [91], pansharpening neural network (PNN) [69], advanced PNN using fine-tuning (A-PNN-FT) [74], and pansharpening by combining ML and traditional fusion schemes (FusionNet) [92], for a fair and critical comparison under the same training and testing data. It is worth to be remarked that we did not select unsupervised learning or GAN-based methods for comparison purposes since they can require different training datasets (with respect to the use ones) invalidating the fair comparison.

A. PanNet

In [73], Yang *et al.* design a deep CNN architecture, called PanNet, for the task of pansharpening relying on the high-frequency information inputs from LRMS and PAN images. The given PanNet architecture considers domain-specific knowledge and mainly focuses on preserving spectral and spatial information in remote sensing images. Overall, the network architecture of the PanNet upsamples first the LRMS image to the PAN scale aiming to keep the spectral information. Besides, a deep residual network is employed to learn spatial mapping to get the spatial details for the fused image. Specifically, the deep residual network mainly contains a pre-processing convolutional layer that increases the feature channels and a post-processing convolutional layer that reduces the channels to the spectral bands. Furthermore, four ResNet blocks [120] with a skip connection are employed to deepen the network depth for a better feature extraction. Especially, the high-frequency spatial information of LRMS and PAN images, which is obtained by using simple high-pass filters, is concatenated and exploited into the deep residual network for its training. With this step, we can learn accurate spatial details that will be added to the LRMS to yield the final HRMS product. The output of the network is then compared with the GT image using an ℓ_2 loss function. By Adam optimizer with momentum, the weights on all the layers can be suitably updated. This strategy focusing on high-frequency content is valid, even getting a good generalization ability. The details about the PanNet (including architecture, hyper-parameter setting, and so forth) are described in Fig. 3.

The idea of the PanNet is to design the network architecture on the high-pass domain rather than the image domain that is commonly used for most of DL-based techniques. The domain-specific high-pass strategy can foster the network generalization capability since images obtained from different sensors have similar distributions for high-frequency information. Also, since most of high-pass details are close to zero, there is a reduction of the mapping space leading to an easier training of the network. In summary, the PanNet demonstrates that the training and generalization abilities of a network can be improved focusing on a specific domain, *i.e.*, the high-pass domain, instead of the original one.

B. DRPNN

Wei *et al.* [71] proposed a deep residual neural network named DRPNN to address the task of pansharpening as shown in Fig. 4. They believed that a deeper CNN with more filtering layers tends to extract more abstract and representative features, and thus a higher prediction accuracy is expected. However, due to the gradient vanishing problem, weights of shallow layers cannot be optimized via backpropagation, which prevents the deep network from being fully learned. Deep residual learning [120] is an advanced method for solving this problem, in which the transformation $\mathcal{F}(\mathbf{X}) \approx CNN(\mathbf{X})$ is replaced with $\mathcal{F}(\mathbf{X}) - \mathbf{X} \approx RES(\mathbf{X})$ by setting a skip connection between the separate layers, which allows them to add more layers to the network and boost its performance. In DRPNN, they built a deep residual skip before and after the convolutional filtering framework contains ten layers with all the kernel sizes set to 7 × 7. Multispectral bands to be fused are interpolated to the PAN scale and then concatenated with the PAN image to form an input cube. After the deep residual feature extraction, a restoration layer with N groups of convolutional filters is employed to obtain the fused images. The outcome is used to calculate the ℓ_2 loss with the GT, and then the stochastic gradient descent (SGD) algorithm is utilized to train the DRPNN, which costs 300 epochs. Besides, they set different learning rates for the first ten layers and the

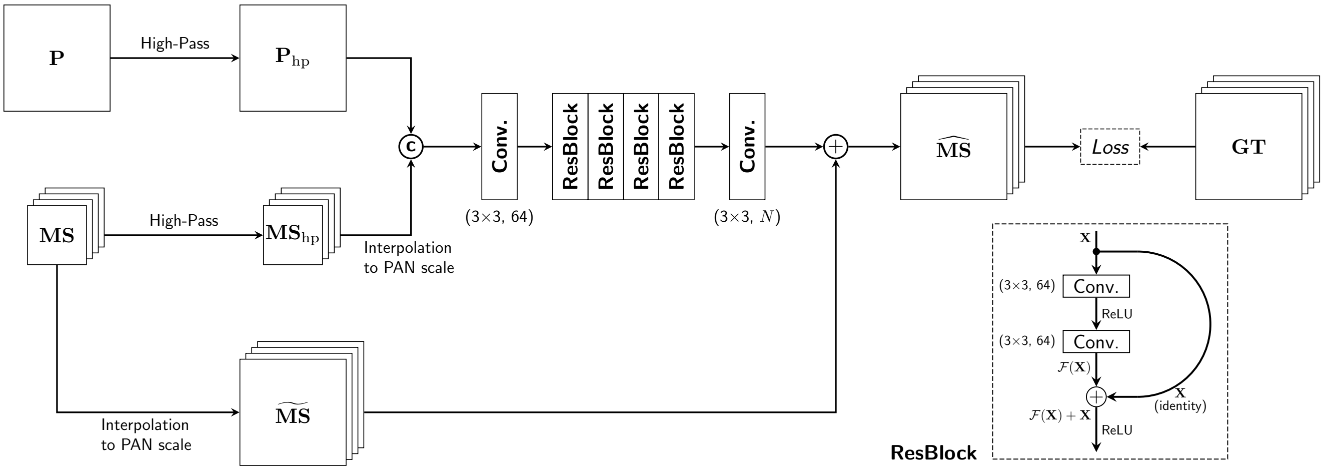


Fig. 3: Flowchart of the PanNet exploiting the ℓ_2 loss function. Note that $(3 \times 3, 64)$ means that the size of the convolutional kernel is 3×3 with 64 channels, and ReLU is the activation function, *i.e.,* rectified linear unit. The notations ⓒ and ⊕ stand for the concatenation and summation, respectively. The $\mathbf{P}_{hp}$ and $\mathbf{MS}_{hp}$ are the high-pass filtered versions of the $\mathbf{P}$ and $\mathbf{MS}$ images, respectively. The **Conv.** block represents a convolutional layer. The upsampling is done using a 23-tap polynomial interpolator [121]. The definitions and notations in the subsequent network architectures are the same as those ones, thus we will not introduce them once again.

last layer, which are 0.05 and 0.005, respectively, while the momentum is fixed at 0.95. Note that after every 60 epochs, the learning rate would fall by half.

The deep residual skip ensures that the model learns the difference between input and output, leading to quick and accurate training. The strategy of the skip connection is also used in the PanNet, published in the same period as DRPNN. DRPNN can get competitive outcomes thanks to its usage of convolution kernels with larger size, *i.e.,* 7×7, which can cover a larger area. However, due to these larger kernels, DRPNN has a relative high parameter amount.

C. MSDCNN

In [75], Yuan *et al.* proposed a multiscale and multi-depth CNN, called MSDCNN, for pansharpening. As shown in Fig. 5, MSDCNN extracts deep and shallow features using different convolutional filters with receptive fields of multiple scales and finally integrates them to yield a better estimation. In pansharpening, the coarse structures and texture details are both of great importance for ideal restoration. At the same time, the sizes of the ground objects vary from very small neighborhoods to large regions containing thousands of pixels, and a ground scene can cover many objects with various sizes. Recalling that multiscale features differently response to convolutional filters with different sizes, they proposed a multiscale block containing three parallel convolutional layers with kernel sizes of 3, 5, and 7. Furthermore, they employed a short skip connection for each multiscale block, which forms the multiscale residual block (MSResB in Fig. 5). Passing the input image cube through the deep extraction branch, the deep features $\mathbf{CNN}_d$ can be extracted, which have been reduced to the same spectral dimensionality as the ideal multispectral images. On the other hand, the shallow features $\mathbf{CNN}_s$ are yielded by a shallow network branch with three convolutional layers, where the kernel sizes are 9, 1, and 5, respectively. Furthermore, the output feature numbers of the convolutional layers in both the branches are reduced as the depth increases. The MSDCNN is also trained for 300 epochs using the ℓ_2 loss function with the SGD optimization algorithm, where the momentum μ is equal to 0.9 and the learning rate ϵ is 0.1.

Overall, the MSDCNN benefits from several features obtained by convolving one feature with kernels having different sizes (called multiscaled operation). By this strategy, different features with various receptive fields are concatenated to improve the feature extraction. Beyond the multiscaled operation in the so-called deep branch, the other branch conducts three plain convolutions to get the so-called shallow features. We think the plain convolution layers in the shallow branch could not be necessary since they make the network outputs from the two branches too flexible, resulting in an uncertainty in learning deep and shallow features.

D. BDPN

In traditional MRA-based pansharpening methods, multiscale details of the PAN image are used to improve the resolution of the MS image. The accuracy of multiscale details is directly related to the quality of the pansharpened image. Insufficient details lead to blurring effects. Instead, excessive details result in artifacts and spectral distortions. To more accurately extract the multiscale details of the HRMS image, Zhang *et al.* [79] propose a two-stream network for pansharpening, which is called BDPN. The network adopts a bidirectional pyramid structure to separately process the MS image and the PAN image, following the general idea of multiresolution analysis. Multilevel details are extracted from the PAN image and injected into the MS image to reconstruct the pansharpened image. The detail extraction branch uses stacked ResBlocks to extract details while the image reconstruction branch uses subpixel convolutional layers to upsample the MS image. The multiscale structure helps the network to extract

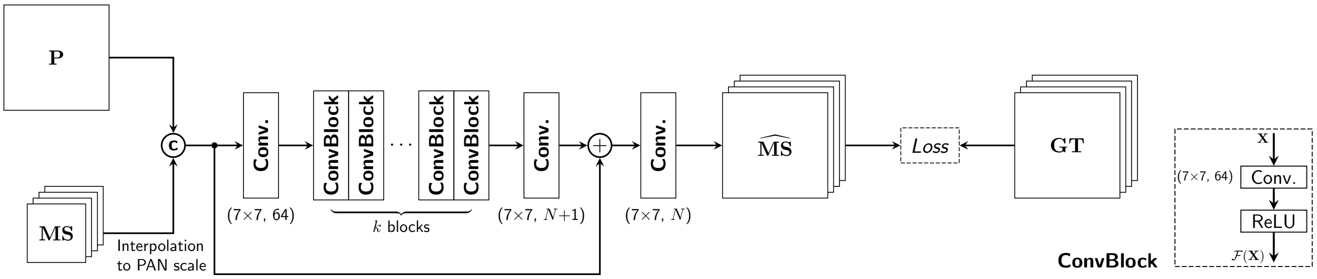


Fig. 4: Flowchart of the DRPNN using the ℓ_2 loss function.

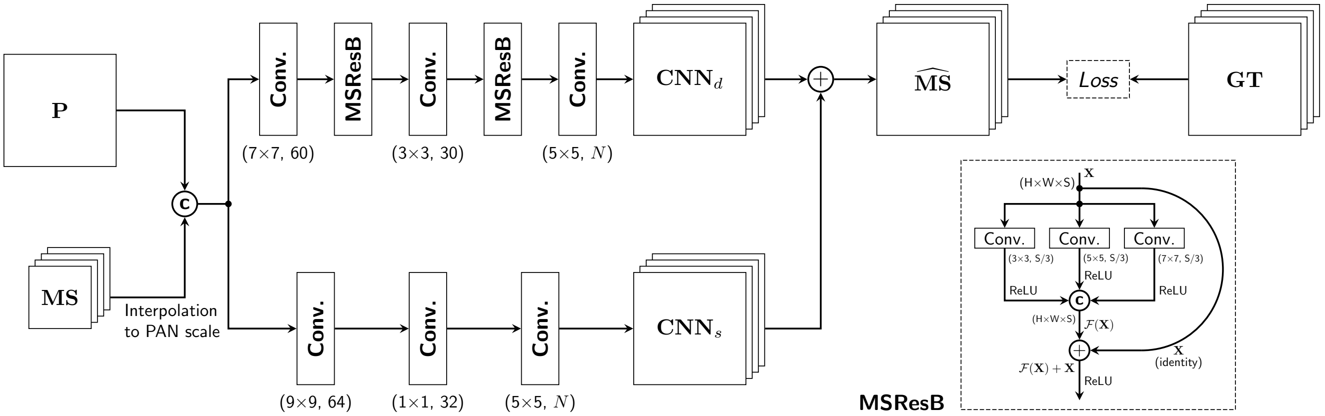


Fig. 5: Flowchart of the MSDCNN using the ℓ_2 loss function, in which MSResB is the multi-scale residual block. Besides, $\mathbf{CNN}_d$ and $\mathbf{CNN}_s$ stand for deep and shallow features, respectively.

multiscale details from the PAN image. It allows part of the computation to be located at reduced resolution features, thus reducing the computation burden. In the network's training, a multiscale loss function is used to accelerate the rate of convergence. At the beginning, reconstructed images at all the scales are supervised. As the training continues, the weight of the low resolution scales gradually declines. Readers can find the detailed flowchart of the BDPN in Fig. 6.

Although the idea of a bidirectional structure has been proposed in other multi-resolution fusion applications, such as, the deep image super-resolution (SR) [122], the BDPN used it first for pansharpening. However, because of the usage of too many multiscaled convolution layers, the BDPN has a large number of parameters, similarly to the DRPNN. This disadvantage can be alleviated by exploiting more effective convolutions.

E. DiCNN

He *et al.* [91] proposed a new detail injection procedure based on DL end-to-end architectures to learn the MS details whilst enhancing the physical interpretability of the pansharpening process. Two detail injection-based CNN models are implemented following the three-layer architecture for super-resolution proposed by Dong *et al.* [90]. Fig. 7 provides a graphical overview of the network used in this work based on the first proposed model in [91].

The adopted DiCNN receives as input the concatenation along the spectral dimension of the PAN image, $\mathbf{P}$, and the MS image upsampled to the PAN scale, $\widetilde{\mathbf{MS}}$. As a result, the volume $\mathbf{G} \in \mathbb{R}^{H\times W\times N+1} = (\widetilde{\mathbf{MS}}, \mathbf{P})$ is obtained as input, where $H \times W$ indicates the spatial dimensions and N the number of spectral bands of the MS plus the PAN image. This input volume $\mathbf{G}$ is processed by a stack of three 3×3 convolution layers, where the first and second layers are followed by the non-linear activation function ReLU to explore the non-linearities of the data. In this regard, the stack of convolution layers exploits the relations between the upsampled MS and PAN images to obtain those MS details that can enhance the original MS data, involving the mapping function $\widehat{\mathbf{D}}(\mathbf{G};\boldsymbol{\theta})$ that obtains the details of the MS fused image from the inputs $\mathbf{G}$ with $\boldsymbol{\theta}$ representing the set of the learnable parameters of the convolutions. Moreover, the DiCNN employs residual learning to enhance the feature extraction process by propagating only the upsampled MS image through a shortcut connection. This not only maintains the same number of spectral bands between the shortcut data and the obtained details (avoiding the implementation of an auxiliary convolution within the shortcut), but it also provides an explicit physical interpretation. Indeed, in contrast to other deep models that work as black boxes, the DiCNN introduces a domain-specific structure with a meaningful interpretation. As a result, the output $\widehat{\mathbf{D}}(\mathbf{G};\boldsymbol{\theta})$ can be directly exploited to enhance the upsampled MS image to produce the desired HRMS image. In this sense, the main goal of the DiCNN is to minimize the loss function $l(\boldsymbol{\theta})$ defined by (5), with the aim of appropriately adjusting the network parameters that best fit

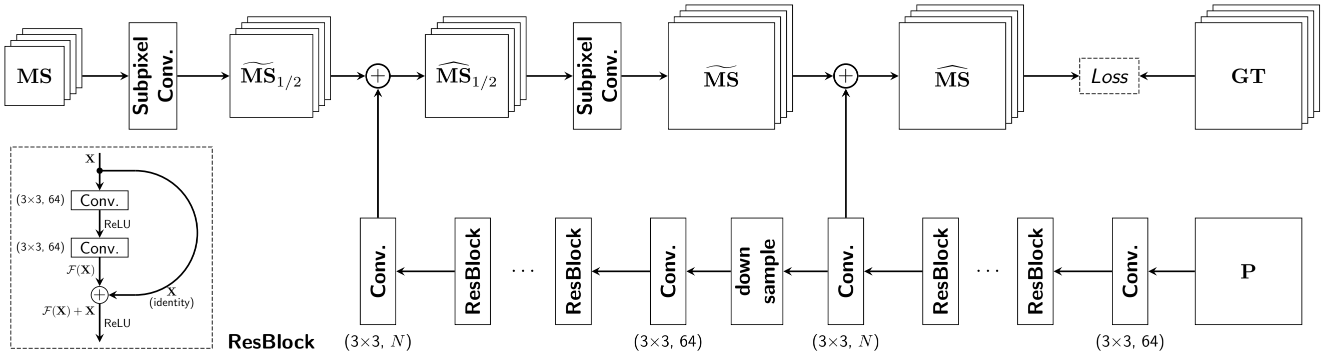


Fig. 6: Flowchart of the BDPN exploiting the Charbonnier loss function, where "**down sample**" means the reduction of the spatial resolution by a factor 2 and "1/2" stands for the upsampling of a factor 2. Finally, the **Subpixel Conv.** block represents a subpixel convolutional layer used to upsample the MS image.

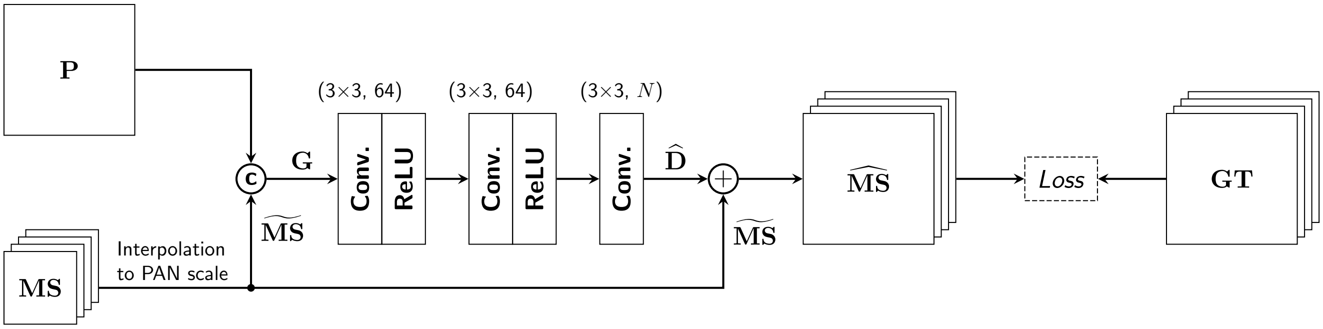


Fig. 7: Flowchart of the DiCNN exploiting the Frobenius loss function.

the data:

$$\begin{aligned} l(\boldsymbol{\theta}) &= \| \widehat{\mathbf{D}}(\mathbf{G};\boldsymbol{\theta}) + \widetilde{\mathbf{MS}} - \mathbf{Y} \|_F^2 \\ &= \frac{1}{N_p}\sum_{i=1}^{N_p} \| \widehat{\mathbf{D}}^{(i)}(\mathbf{G}^{(i)};\boldsymbol{\theta}) + \widetilde{\mathbf{MS}}^{(i)} - \mathbf{Y}^{(i)} \|_F^2, \end{aligned} \tag{5}$$

where $\mathbf{Y}$ denotes the ground-truth image, N_p is the number of the total input patches, i is the index of the current patch, and $\| \cdot \|_F$ is the Frobenius norm. This guarantees that the DiCNN approach can directly learn the details of the MS data.

Overall, the strategy of the skip connection, as for PanNet, DRPNN, and MSDCNN, is employed again in the DiCNN to have a fast convergence with an accurate computation. Thanks to the use of only three convolution layers, DiCNN involves significantly fewer network parameters (see also Tab. XIX), in the meanwhile holding competitive pansharpened performance.

F. PNN

The pansharpening convolutional neural network model by Masi *et al.* [69], called PNN, is among the first pansharpening solutions based on CNNs. Inspired by the super-resolution network for natural images proposed in [90], PNN is a very simple three-layer fully-convolutional model. Tab. III reports the main hyper-parameters related to the PNN implementation for the proposed toolbox where, differently from the original version, they have been set equal for all sensors, with the obvious exception for the number of input and output channels of the whole network that are related to the number of the spectral bands of the MS image. The three convolutional layers are interleaved by rectified linear unit (ReLU) activations. Prior to feed the network, the input MS component is upscaled to the PAN size via 23-tap polynomial interpolation and concatenated with the PAN to form a single input data cube.

Although the PNN exploits the CNN architecture for single image SR in [90], just extending it to the pansharpening task, this approach holds a quite important role in the DL-based pansharpening community. In fact, it is the first attempt to address the pansharpening issue using a fully convolutional neural network, resulting in an important benchmark for subsequently developed DL-based pansharpening techniques. Since the main structure of the PNN only involves three simple convolution layers without any skip connection, its parameters are not significant getting a relatively slow convergence.

G. A-PNN-FT

Two years later, Scarpa *et al.* [74] proposed an advanced version of PNN which presents three main innovations: residual learning, ℓ_1-loss, and a fine-tuning for target adaptation. Hereinafter, this solution will be referred to as A-PNN-FT. Residual learning [120] is an important innovation in deep learning, introduced with the primary purpose of speeding-up the training process for very deep networks, as it helps preventing vanishing gradient problems. However, it has soon demonstrated to be a natural choice for resolution enhancement [123]. In fact, the desired super-resolved image can be viewed as composed of its low and high-frequency components, the former being essentially the input low-resolution image, the latter being the missing (or residual) part to be actually restored. Residual schemes naturally address super-resolution or pansharpening problems in light of this partition, avoiding the unnecessary reconstruction of the whole desired output and reducing the risk of altering the low-frequency content of the image (*i.e.,* spectral distortion). As a matter of fact, the majority of the recent DL pansharpening models embed residual modules [71], [73], [74], [76], [78], [79]. Specifically for A-PNN/-FT, a single input-output skip connection added to the PNN model converts it in a global residual module as highlighted by the semitransparent blocks of Fig. 8, where it is summarized the overall A-PNN-FT algorithm. Solid line connections refer to the fine-tuning phase. Differently from the usual training where data samples do not come from the test images, in fine tuning the same test image is used for parameters update as shown in figure. This makes perfectly sense thanks to the self-supervised learning allowed by the resolution downgrade process that generates labeled samples from the input itself. Further details about the training (pre-training for A-PNN-FT) of all the toolbox models will be later provided in a dedicated section. When fine-tuning starts, the model parameters Φ_0 correspond to those computed in pre-training and they are associated to what is referred to as A-PNN. After a prefixed number of tuning iterations (50 by default) on the target (rescaled) test image, the parameters are frozen (say Φ_∞) and eventually (follow dashed lines) the full-resolution test image can be pansharpened using the "refined" A-PNN, that is A-PNN-FT.

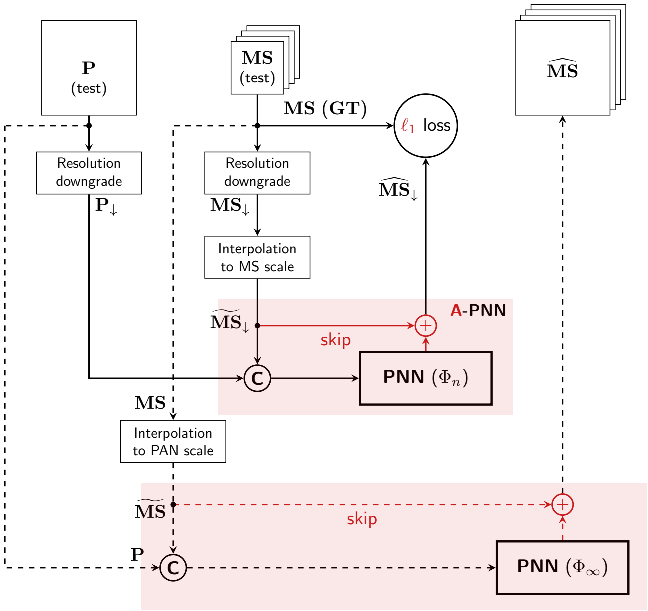


Fig. 8: Top-level flowchart of the A-PNN-FT. Reduced resolution adaptation (solid lines) and full resolution test (dashed) phases. The core A-PNN model is highlighted by opaque blocks and differs from PNN for the introduction of a skip connection (red lines) and the ℓ_1 loss which replaces an ℓ_2. The symbol $\downarrow$ indicates a resolution downgraded version of the image where applied.

H. FusionNet

The traditional approaches such as CS and MRA have achieved competitive outcomes in pansharpening. Nevertheless, these traditional methods are under the assumption of linear injection models, which can be unsuitable according to the real spectral responses for sensors typically used in pansharpening. This motivates utilizing nonlinear approaches such as ML to avoid the limitation of the above-mentioned linear injection models. In [92], Deng *et al.* exploit the combination of ML techniques and traditional fusion schemes, *i.e.,* CS and MRA, to address the task of pansharpening. The overall network architecture, called FusionNet, estimates the nonlinear injection models that rule the combination of the upsampled LRMS image and the extracted details exploiting the two philosophies. In particular, the extracted details can be calculated by directly inputting the difference between the duplicated PAN image and the upsampled LRMS image into a deep residual network. This strategy of directly differencing the duplicated PAN and the upsampled LRMS images is simple. However, it can preserve the latent spatial and spectral information from PAN and LRMS images, respectively. Besides, the extracted details are taken into account in a pre-processing convolutional layer to increase the feature channels and then passing four ResNet blocks to deepen the network depth for a better feature extraction. The generated features are convoluted by a post-processing layer to get the HRMS details consisting of the same LRMS spectral band number. Moreover, the learned HRMS details are directly added to the upsampled LRMS to yield the HRMS outcome. FusionNet exploits an Adam optimizer with momentum and a fixed learning rate to train the network. The conventional ℓ_2 function is selected as loss function to measure the distance between the HRMS outcome and the GT image. Readers can refer to Fig. 9 for further details about the FusionNet approach.

Thanks to the combination of ML techniques and traditional fusion schemes to design the network architecture, FusionNet can have a better and faster regression between inputs and labels getting competitive results when training and testing datasets have similar structures. However, since FusionNet is also built by plain convolution layers like PNN and DiCNN (even with skip connection), its network generalization is weaker than PanNet and A-PNN-FT, which are based on specific operations, such as, learning in the high-pass domain and fine-tuning.

IV. Experimental Results

This section is devoted to the description of the experimental results. The quality assessment protocols will be briefly detailed together with the datasets and the benchmark used for comparison purposes. Afterwards, the generation of the training data and the parameters tuning will be provided. Finally, the results both at reduced and full resolutions will be summed up including a discussion about computational burden, convergence, and other peculiarities of ML-based approaches.

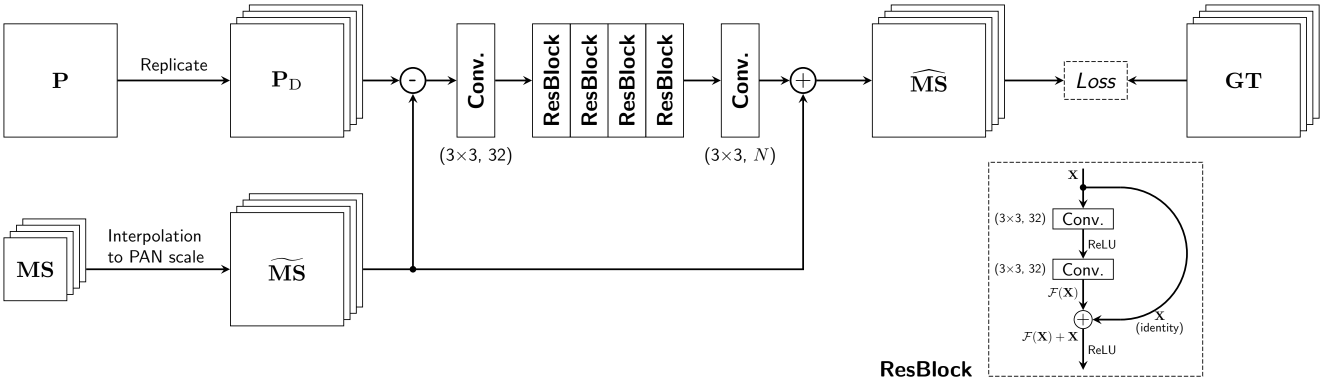


Fig. 9: Flowchart of the FusionNet exploiting the ℓ_2 loss function. $\mathbf{P}_{\mathrm{D}}$ is the replicated version of $\mathbf{P}$ along the spectral dimension.

A. Quality Assessment of Fusion Products

Quality assessment of pansharpening methods and data products is a very debated problem. Wald's protocol [124] gives an answer to this issue by introducing two main properties (*i.e.,* consistency and synthesis) that a fused product should satisfy.

To verify the synthesis property, a reduced resolution assessment is considered. Thus, the original MS and PAN images are degrading by spatially filtering them to a reduced resolution. Then, the pansharpening algorithm is applied to these data and the outcome is compared with the original MS data playing the role of the reference image. The more the similarity between the fused and the reference images, the higher the performance of the pansharpening approach. Clearly, the choice of the filters to spatially degrade the MS and PAN products could bias the assessment. Generally, spatial filters matching the MS sensor's modulation transfer functions (MTFs) are exploited to degrade the MS image. Instead, ideal filters are adopted to reduce the resolution of the PAN image [4]. The similarity between the fused and the reference images is measured by exploiting the following multidimensional indexes: the spectral angle mapper (SAM) [125], the relative dimensionless global error in synthesis (ERGAS) [126], and the multi-band extension of the universal image quality index, Q2^n [127]. The ideal results are 0 for SAM and ERGAS and 1 for Q2^n.

Unfortunately, the sole reduced resolution assessment is not enough to state the superiority of a pansharpening algorithm. Indeed, an implicit hypothesis of "invariance among scales" is performed when working at reduced resolution. Thus, even though this assessment is very accurate, it is based on the validity of the above-mentioned assumption. To this aim, the full resolution assessment is also considered. In this case, no hypothesis is done, but the lack of a reference image reduces the accuracy in the performance assessment. In this paper, the hybrid quality with no reference (HQNR) index is used. This latter borrows the spatial distortion index, D_S, from the quality with no reference (QNR) [128], and the spectral distortion index, D_λ, from Khan's protocol [129]. The two distortions are combined as follows:

$$\text{HQNR} = (1 - D_\lambda)^\alpha \; (1 - D_S)^\beta, \tag{6}$$

where $\alpha = \beta = 1$. Ideal values for the D_S and the D_λ indexes are 0, thus we have that the optimal value for the HQNR is 1.

B. Datasets

Several different test cases acquired by five widely used sensors for pansharpening are considered. For all the sensors, both the assessment at reduced resolution (RR) and at full resolution (FR), following the indications drawn in Sect. IV-A, are provided. The characteristics of the employed datasets are detailed as follows.

a) WorldView-2 datasets: These data were acquired by the WorldView-2 (WV2) sensor, which works in the visible and near infrared spectrum range. The multispectral sensor is characterized by eight spectral bands (coastal, blue, green, yellow, red, red edge, NIR1, and NIR2), and also, a panchromatic channel is available. The spatial sampling interval (SSI) is 1.84 m for MS and 0.46 m for PAN, respectively. The resolution ratio R is equal to 4. The radiometric resolution is 11 bits.

Three datasets are exploited: i) WV2 Washington representing a mixed area in the surroundings of Washington, USA, characterized by an elevate presence of high buildings, vegetated areas, and a river (the size of an MS spectral band is 6248×5964), see Fig. 10; ii) WV2 Stockholm depicts a mixed zone showing several water bodies in the urban area of Stockholm, Sweden (the size of an MS spectral band is 1684×2176), see Fig. 10; iii) WV2 Rio instead shows a mixed area of the city of Rio de Janeiro, Brazil, characterized by vegetated and urban features and a small portion of sea in the top right side of the image (the size of an MS spectral band is 512×512), see Fig. 11. i) and ii) are used for training and testing the networks at reduced resolution following Wald's protocol as in Sect. IV-A. Instead, iii) is exploited to test the ability of the networks to work in real conditions, namely with a quite different dataset because of an acquisition of the same sensor but over a different area of the world and in a different time, thus showing different features such as atmospheric conditions, haze, landscapes, solar elevation angle, and so forth. In this latter case, both a reduced and a full resolution assessment are performed.

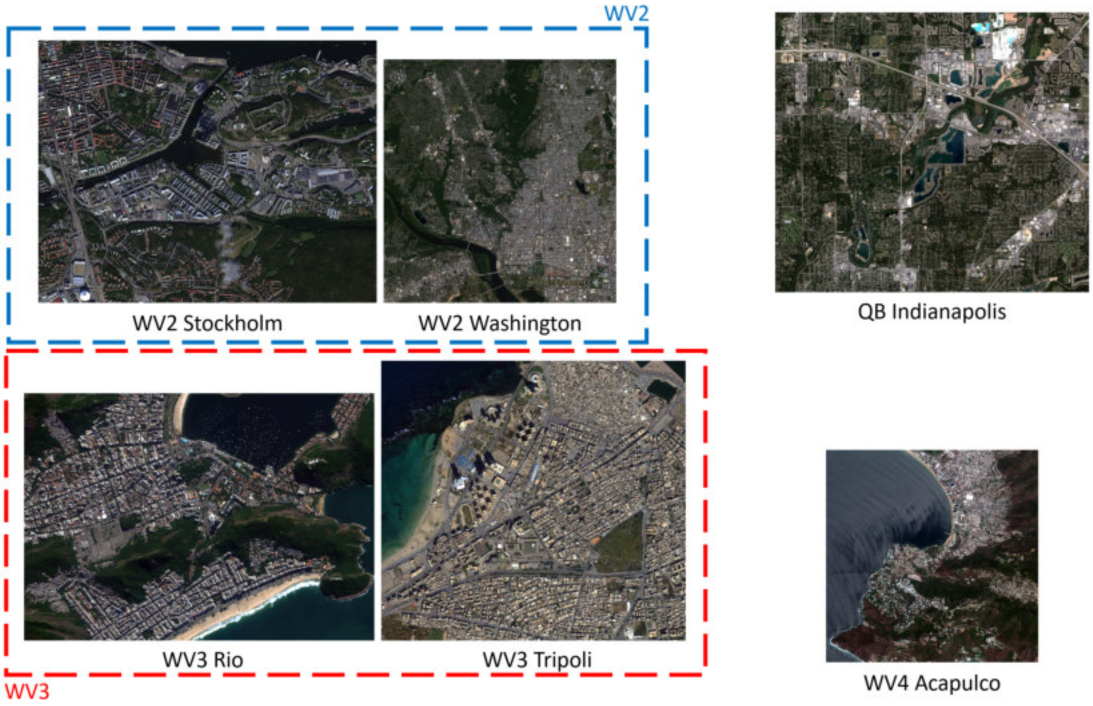


Fig. 10: Datasets used for training ML approaches (selected bands: red, green, and blue). Note that the images related to the datasets are intensity stretched to aid the visual inspection.

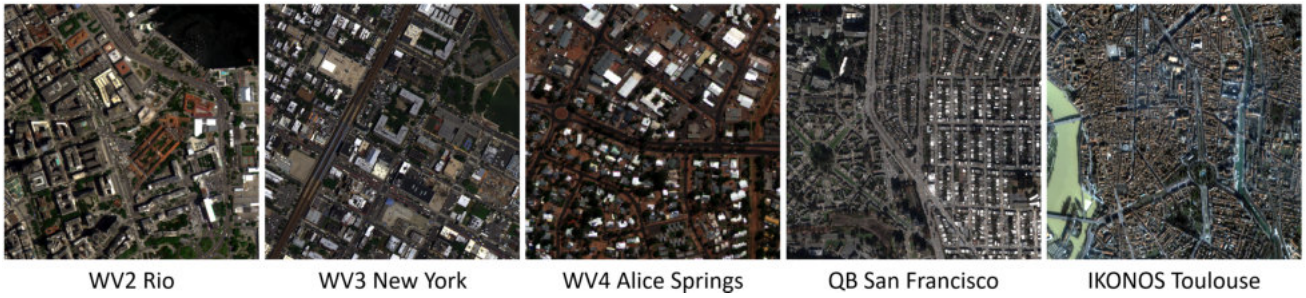


Fig. 11: Datasets used for testing ML approaches (selected bands: red, green, and blue). Note that the images related to the datasets are intensity stretched to aid the visual inspection.

b) WorldView-3 datasets: These data were acquired by the WorldView-3 (WV3) sensor, which works in the visible and near infrared spectrum range. The multispectral sensor is characterized by eight spectral bands (the same as the WV2 MS sensor), and also, a panchromatic channel is available. The SSI is 1.2 m for MS and 0.3 m for PAN, respectively. R is equal to 4. The radiometric resolution is 11 bits.

Three datasets are exploited: i) WV3 Tripoli representing an urban area of Tripoli, a coastal town in Libya (the size of an MS spectral band is 1800×1956), see Fig. 10; ii) WV3 Rio that is a mixed dataset showing both vegetated and man-made structures in the surroundings of Rio de Janeiro, Brazil (the size of an MS spectral band is 2380×3376), see Fig. 10, again; iii) WV3 New York instead depicts an urban area of the city of New York (USA) with a more relevant presence of high buildings with respect to European urban scenarios (the size of an MS spectral band is 512×512), see Fig. 11. Again, i) and ii) are used for training and testing the networks at reduced resolution following Wald's protocol. Instead, the real test cases (both at reduced resolution and at full resolution) are performed by exploiting iii).

c) WorldView-4 datasets: These data were acquired by the WorldView-4 (WV4) sensor, which works in the visible and near infrared spectrum range. The multispectral sensor is characterized by four spectral bands (blue, green, red, and NIR), and also, a panchromatic channel is available. The SSI is 1.24 m for MS and 0.31 m for PAN, respectively. R is equal to 4. The radiometric resolution is 11 bits.

Two datasets are exploited: i) WV4 Acapulco representing a mixed area with sea, land, and urban areas in the surroundings of the city of Acapulco, Mexico (the size of an MS spectral band is 4096×4096), see Fig. 10; ii) WV4 Alice that is a mixed dataset mainly showing urban and bare soil features related to the city of Alice Springs, Australia (the size of an MS spectral band is 512×512), see Fig. 11. Again, i) is used for training and testing the networks at reduced resolution following Wald's protocol. Instead, the real test cases (both at reduced resolution and at full resolution) are performed by exploiting ii).

d) QuickBird datasets: These data were acquired by the QuickBird (QB) sensor, which works in the visible and near infrared spectrum range. The multispectral sensor is characterized by four spectral bands (blue, green, red, and NIR). Also, a panchromatic channel is available. The SSI is 2.44 m for MS and 0.61 m for PAN, respectively. R is equal to 4. The radiometric resolution is 11 bits.

Two datasets are exploited: i) QB Indianapolis representing a mixed area with the high presence of man-made structures, but with also water bodies and green areas captured over the city of Indianapolis, USA (the size of an MS spectral band is 3624×4064), see Fig. 10; ii) QB San Francisco is an urban area of San Francisco, USA (the size of an MS spectral band is 512×512), see Fig. 11. Again, i) is used for training and testing the networks at reduced resolution following Wald's protocol. Instead, the real test cases (both at reduced resolution and at full resolution) are performed by exploiting ii).

e) IKONOS dataset: This dataset represents an urban area of the city of Toulouse, France. It was acquired by the IKONOS sensor, which works in the visible and near infrared spectrum range. The multispectral sensor is characterized by four spectral bands as for the QB sensor, and also a panchromatic channel is available. The resolution cell is 4 m $\times$ 4 m for the MS bands and 1 m $\times$ 1 m for the PAN channel. R is, therefore, equal to 4. The radiometric resolution is 11 bits. The size of an MS spectral band is 512×512 pixels, see Fig. 11. This dataset is used to assess the generalization ability of the networks with respect to the changing of both acquisition sensor and captured scenario. In particular, we exploited networks trained on the QB training set, but showing the performance on this IKONOS dataset.

C. Benchmark

Several state-of-the-art algorithms are employed for comparison purposes:

- EXP: MS image interpolation using a polynomial kernel with 23 coefficients;
- CS methods
  - BT-H: optimized Brovey transform with haze correction [30];
  - BDSD-PC: band-dependent spatial-detail with physical constraints [117];
  - C-GSA: context-based Gram-Schmidt adaptive with local parameter estimation exploiting clustering [16];
- MRA methods
  - MTF-GLP-FS: generalized Laplacian Pyramid (GLP) with MTF-matched filters with a full scale (FS) regression-based injection model [119];
  - MTF-GLP-HPM-R: GLP with MTF-matched filters and high-pass modulation injection model with a preliminary regression-based spectral matching phase [118];
- VO methods
  - SR-D: pansharpening based on sparse representation of injected details [66];
  - TV: pansharpening based on total variation [50];
- ML methods
  - PanNet: deep CNN architecture for pansharpening [73];
  - DRPNN: deep residual neural network for pansharpening [71];
  - MSDCNN: multiscale and multidepth CNN architecture for pansharpening [75];
  - BDPN: bidirectional pansharpening network [79];
  - DiCNN: detail injection based convolutional neural network [91];
  - PNN: pansharpening neural network [69];
  - A-PNN-FT: advanced pansharpening neural network using an adaptive tuning scheme [74];
  - FusionNet: pansharpening by combining ML and traditional fusion schemes [92].

A more detailed description of the methods can be found in Sects. II and III and in the related references.

D. Generation of Training Data

The building of the training set is a crucial step for ML-based pansharpening approaches. Although in the literature there are plenty of state-of-the-art ML-based methods, the way of generating training sets is often different leading to an unfair comparison among them. This section is devoted to the illustration of the whole procedure of generating training samples for ML-based pansharpening. Moreover, the MATLAB code for simulating training sets will be distributed to the community.

The overall procedure of generating training samples is depicted in Fig. 12, which follows three main steps:

1) **Data download**. Because of license limitations, we are not permitted to share the original data. Readers can directly download them from commercial websites[2].
2) **Data simulation**. After downloading the source images, we can read the original PAN and MS images. Afterwards, according to Wald's protocol, we filter the original MS image matching the corresponding sensor's modulation transfer function (MTF)[3] and the original PAN image using an almost ideal filter, then downgrading the filtered images by the nearest neighbor interpolation with a scale factor of 4. Furthermore, the downsampled MS image will be upsampled to the PAN scale by a 23-tap polynomial interpolator. Hence, we will exploit in the training phase: i) the downsampled PAN image; ii) the downsampled MS image; iii) the original MS image as GT; iv) the upsampled version of the downsampled MS image (denoted as UMS from hereon). Refer to Fig. 12 and Tab. II for more details about the simulation process and the data used in this work, respectively.
3) **Data patching**. The simulated images in step 2 are too big (considering the limited storage capabilities

[2] For WorldView data, the interesting readers can refer to https://resources.maxar.com/

[3] The MATLAB code about filtering using MTF can be found at the following link: *The link will be added after the paper acceptance.*

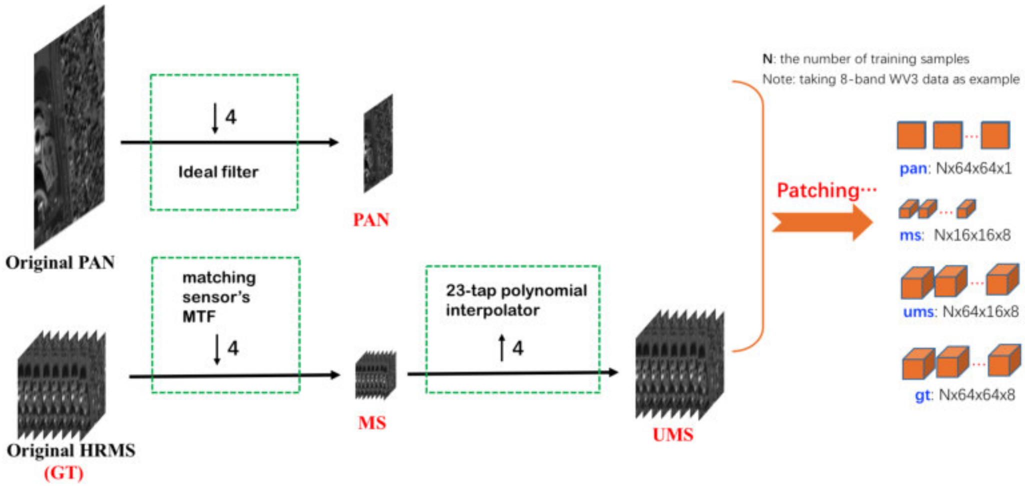
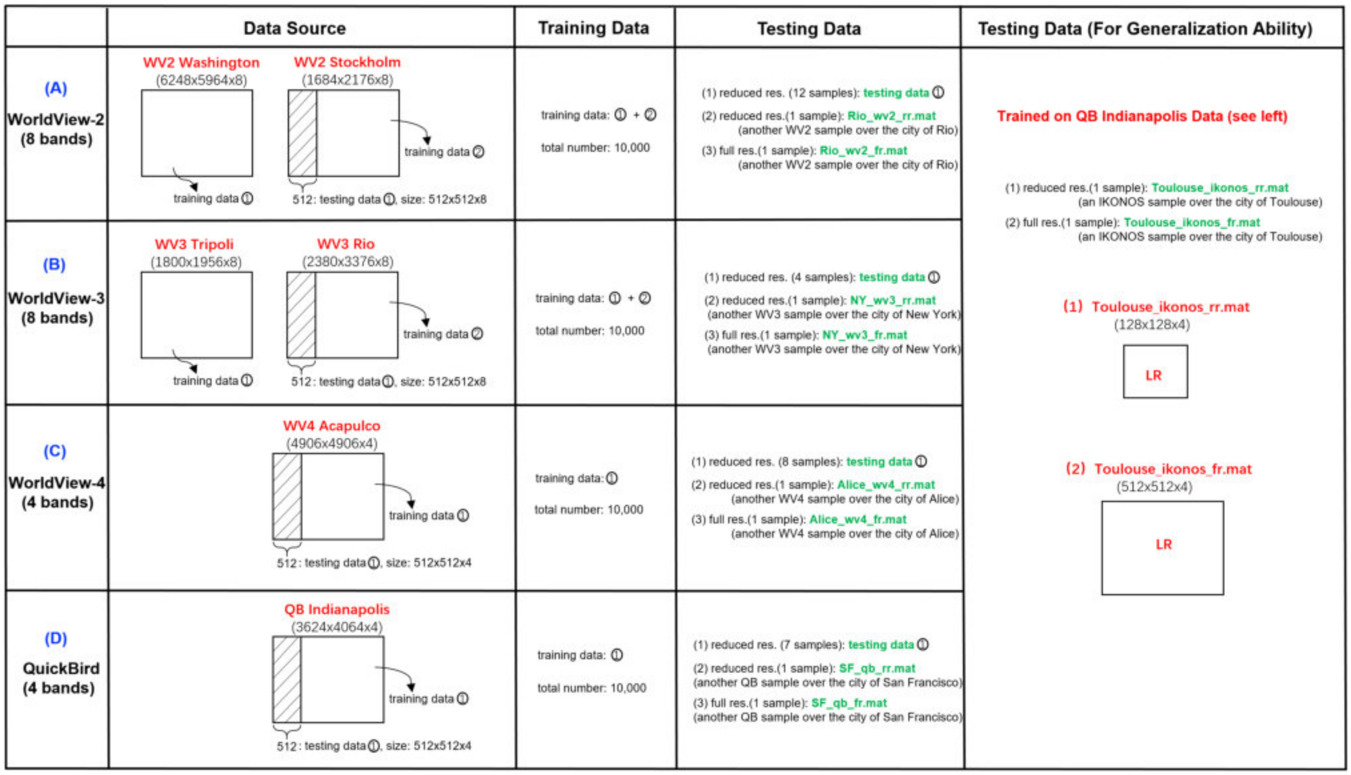


Fig. 12: The generation process of training samples by Wald's protocol. Note that the names highlighted in red are about the generated training data used to feed the networks, *i.e.*, the GT, the low spatial resolution MS image, the PAN, and the upsampled MS image (here denoted as UMS).

TABLE II: Details about the training and testing datasets used in this work (see also Figs. 10 and 11). "res." means resolution.



	Data Source	Training Data	Testing Data	Testing Data (For Generalization Ability)
(A) WorldView-2 (8 bands)	WV2 Washington (6248x5964x8) training data ① WV2 Stockholm (1684x2176x8) training data ② 512: testing data ①, size: 512x512x8	training data: ① + ② total number: 10,000	(1) reduced res. (12 samples): testing data ① (2) reduced res.(1 sample): Rio_wv2_rr.mat (another WV2 sample over the city of Rio) (3) full res.(1 sample): Rio_wv2_fr.mat (another WV2 sample over the city of Rio)	Trained on QB Indianapolis Data (see left) (1) reduced res.(1 sample): Toulouse_ikonos_rr.mat (an IKONOS sample over the city of Toulouse) (2) full res.(1 sample): Toulouse_ikonos_fr.mat (an IKONOS sample over the city of Toulouse) (1) Toulouse_ikonos_rr.mat (128x128x4) LR (2) Toulouse_ikonos_fr.mat (512x512x4) LR
(B) WorldView-3 (8 bands)	WV3 Tripoli (1800x1956x8) training data ① WV3 Rio (2380x3376x8) training data ② 512: testing data ①, size: 512x512x8	training data: ① + ② total number: 10,000	(1) reduced res. (4 samples): testing data ① (2) reduced res.(1 sample): NY_wv3_rr.mat (another WV3 sample over the city of New York) (3) full res.(1 sample): NY_wv3_fr.mat (another WV3 sample over the city of New York)	
(C) WorldView-4 (4 bands)	WV4 Acapulco (4906x4906x4) training data ① 512: testing data ①, size: 512x512x4	training data: ① total number: 10,000	(1) reduced res. (8 samples): testing data ① (2) reduced res.(1 sample): Alice_wv4_rr.mat (another WV4 sample over the city of Alice) (3) full res.(1 sample): Alice_wv4_fr.mat (another WV4 sample over the city of Alice)	
(D) QuickBird (4 bands)	QB Indianapolis (3624x4064x4) training data ① 512: testing data ①, size: 512x512x4	training data: ① total number: 10,000	(1) reduced res. (7 samples): testing data ① (2) reduced res.(1 sample): SF_qb_rr.mat (another QB sample over the city of San Francisco) (3) full res.(1 sample): SF_qb_fr.mat (another QB sample over the city of San Francisco)	

of the graphics processing units, GPUs) to feed the pansharpening networks. Thus, we need to crop these simulated images in the step 2 into small patches. Hence, we segment the GT, the UMS, the PAN, and the MS images into several small patches with sizes $64 \times 64 \times 8$ (with an overlap of 16 spatial pixels), $64 \times 64 \times 8$ (with an overlap of 16 spatial pixels), $64 \times 64 \times 1$ (with an overlap of 16 spatial pixels), and $16 \times 16 \times 8$ (correspondingly, with an overlap of 4 spatial pixels due to the scale factor of 4), respectively[4]. We finally have 9000 training samples (*i.e.*, 9000 patch images) and 1000 validation samples for WV3, WV4 and QB datasets, and 14496 training samples and 1611 validation samples for the WV2 dataset, which can avoid overfitting during the training phase. Please, refer to Fig. 12 for more details, again.

[4] The MATLAB code for patching the training datasets can be found at the following link: *The link will be added after the paper acceptance.*

TABLE III: Optimal parameters for the eight compared ML-based methods. Notation: **Epo. #** (epoch number), **Bs** (mini-batch size), **Algo** (optimization algorithm), **Initi. Lr** (initial learning rate), **Lr Tun.** (tuning strategy of learning rate), **Fs** (filter size for each layer), **Filt. #** (filter number for each layer), **Loss Ty.** (type of loss function), and **Ly. #** (number of layers). The WV3 dataset is here used as an exemplary case. The training dataset consists of 10000 WV3 samples with size $64 \times 64 \times 8$.

Para.	**PNN**	**A-PNN-FT**	**DRPNN**	**MSDCNN**	**PanNet**	**DiCNN**	**BDPN**	**FusionNet**
Epo. #	12,000	10,000	500	500	450	1000	1,000	400
Bs	64	64	64	64	32	64	8	32
Algo	SGD	SGD	Adam	Adam	Adam	Adam	Adam	Adam
Initi. Lr	0.0289*bands	0.0289*bands	2×10^{-4}	1×10^{-4}	0.001	2×10^{-4}	0.0001	0.0003
Lr Tun.	fixed initi. Lr (FIL)	FIL	×0.5 per 60 Epos.	×0.5 per 60 Epos.	FIL	×0.5 per 200 Epos.	×0.8 per 100 Epos.	FIL
Fs	9×9, 5×5	9×9, 5×5	3×3	3×3	3×3	3×3	3×3	3×3
Filt. #	64, 32	64, 32	32	32	64	32	64	32
Loss Ty.	ℓ_2	ℓ_1	ℓ_2	ℓ_2	ℓ_2	Frobenius	Charbonnier	ℓ_2
Ly. #	3	3	11	12	10	3	43	10

E. Parameters Tuning

This section shows the parameter settings for all the compared ML-based pansharpening methods, including information about epoch number, learning rate, optimization algorithm, loss function, and so forth. These details can be found in Tab. III. Note that some ML-based methods were originally implemented on other platforms (e.g., Tensorflow or MatConv). When we migrated the codes to Pytorch, the original parameters have been tuned again accounting for a different behavior of build-in functions (e.g., a different weight initialization) in the adopted software platform.

F. Assessment on WV2 Data

In this section, we will analyze the outcomes obtained on some WV2 test cases. Multiple reduced resolution testing datasets are evaluated first. Then, another dataset is used to assess the performance both at reduced resolution and at full resolution.

1) Performance on 12 Reduced Resolution Testing Datasets: We evaluate first the quantitative performance of all the compared pansharpening methods on 12 WV2 reduced resolution testing datasets acquired over Stockholm, *i.e.,* the testing data in Tab. II (A) (see also the WV2 Stockholm dataset in Fig. 10). Note that the multiple testing samples are captured over a similar area at same time as those of the training dataset, but exploiting different cuts. By looking at the quantitative performance displayed in Tab. IV, it is easy to see that the ML-based approaches get better average indicators than those of traditional techniques, also showing a smaller standard derivations (*stds*) indicating a better robustness. Specifically, FusionNet outperforms the other compared approaches on these testing data. Besides, PanNet, DRPNN, and DiCNN also have competitive performance. Overall, because the training dataset has similar properties as those of the testing samples, the outcomes of ML-based methods show a clear superiority with respect to traditional techniques. This corroborates the ability of the networks during the training phase to properly fit their weights, thus easily solving similar problems as the ones proposed in this testing phase.

TABLE IV: Average results for the approaches belonging to the benchmark on the reduced resolution WV2 Stockholm testing dataset, *i.e.*, on the 12 WV2 testing datasets in Tab. II (A). Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q8 (± std)	**SAM** (± std)	**ERGAS** (± std)
	CS/MRA/VO		
GT	1.0000±0.0000	0.0000±0.0000	0.0000±0.0000
EXP	0.5812±0.0569	7.4936±1.2394	7.0288±0.8265
BT-H	0.8501±0.0410	6.5042±1.3519	4.1552±0.5579
BDSD-PC	0.8430±0.0477	7.1664±1.2654	4.3242±0.5203
C-GSA	0.8323±0.0442	7.8657±1.3074	4.6591±0.4554
SR-D	0.8321±0.0457	6.6042±1.3383	4.3915±0.6267
MTF-GLP-HPM-R	0.8356±0.0446	7.3204±2.0298	5.0992±2.3204
MTF-GLP-FS	0.8347±0.0391	7.4497±1.6581	4.5257±0.6078
TV	0.7940±0.0834	7.2902±0.9685	4.8400±0.3226
	ML		
PanNet	0.9130±0.0551	4.4143±0.6642	2.7713±0.3156
DRPNN	0.9109±0.0528	4.4730±0.6906	2.8552±0.3393
MSDCNN	0.9079±0.0540	4.5698±0.7250	2.9078±0.3469
BDPN	0.8924±0.0578	5.1381±0.8587	3.2144±0.3781
DiCNN	0.9111±0.0528	4.4857±0.7061	2.8411±0.3365
PNN	0.9043±0.0573	4.6774±0.7064	2.9374±0.3369
A-PNN-FT	0.8991±0.0519	4.9263±0.8348	3.1363±0.3887
FusionNet	**0.9169±0.0532**	**4.2632±0.6336**	**2.6911±0.3115**

2) Performance on the Reduced Resolution WV2 Rio Dataset: This section evaluates the performance of all the compared methods on a single reduced resolution WV2 test case. Differently from the reduced resolution WV2 Stockholm testing samples in Sect. IV-F1, in this case, the single WV2 testing dataset is acquired in a different time over the city of Rio, which represents another area of the world with respect to the ones of the training set. Readers can have a look at the WV2 Rio testing image in Fig. 11. Specifically, Fig. 13 depicts that most of the traditional methods have a better visual appealing than ML-based approaches. Generally speaking, only small differences among the compared techniques can be identified. One exception is represented by the outcome provided by the PNN that has a high spectral distortion. A-PNN-FT (which is an extension of the PNN) instead gets competitive visual performance with a high spectral preservation, thus demonstrating the effectiveness of using the fine-tuning strategy for PNN-based approaches.

The quantitative results are reported in Tab. V. From the table, the differences among all the compared methods are easily remarked. Most of the traditional state-of-the-art methods achieve very high indicators, thus demonstrating their spatio-spectral preservation ability. BT-H method gets the highest Q8 indicator and MTF-GLP-HPM-R obtains the lowest ERGAS among all the compared methods. In contrast, we can observe that the results of the ML-based methods are quite different among each other. Some ML-based approaches, *e.g.,* PanNet, A-PNN-FT, DRPNN, MSDCNN, still have promising results, whereas some other methods, such as PNN, BDPNN, and DiCNN, get a significant reduction of the performance. The possible reason for this phenomenon is that the testing dataset used here is quite different with respect to the training data, *e.g.,* different acquiring area and time. Generally speaking, the more the parameters to be trained, the greater the amount of the data required to estimate them. Moreover, to improve the generalization ability, the training set should consist of samples acquired in several areas and in different conditions to allow to pass to the network a complete knowledge of the problem at hand. In absence of a huge and variegated training set, this kind of analysis will reward just networks designed with a higher generalization ability or networks, such as, the A-PNN-FT, which exploit mechanisms like the fine-tuning that allows the adaptation of the network to the specific problem presented during the testing phase. Thus, A-PNN-FT gets the lowest SAM metric, which indicates the better spectral preservation. Overall, among all the ML-based methods, PanNet gets promising outcomes, but its performance is still lower than most of the compared traditional approaches. More conclusions about the compared ML-based methods can be found seeing the best ML-based methods underlined in Tab. V for each quality metric.

3) Performance on the Full Resolution WV2 Rio Dataset: Apart from the evaluation of the reduced resolution datasets, an assessment at full resolution is also needed. To this aim, an original full resolution WV2 Rio dataset is used, see Fig. 11. Note that the full resolution WV2 Rio data is also acquired over a different area and in a different time comparing it with the WV2 training data. Especially, since there is no GT image, we exploit proper indexes with no reference. We selected the HQNR (consisting in the combination of D_λ and D_S) to have a quantitative evaluation of the performance, as introduced in Sect. IV-A. Tab. VI reports the quantitative results. It is easy to observe that most of traditional state-of-the-art approaches get high performance (even comparing them with the ones of the ML-based methods). In particular, two traditional methods, *i.e.,* SR-D and TV, rank first and third among all the sixteen compared techniques. Instead, the PanNet is the best ML-based approach and gets the second position in the overall ranking, showing its good network generalization. The reason could be the network training conducted on only high-frequency information. Moreover, some ML-based approaches yield lower indexes than traditional methods because the ML-based methods are practically trained on different training samples (as underlined in the previous section), but, in this case, also on reduced resolution samples showing data with a lower spatial resolution than the full resolution ones (this is the main drawback of training ML-based approaches in a supervised manner). This conclusion is also referred to the PNN that obtains the worst HQNR among all the techniques, not only because of its relatively small size but most likely for the lack of residual modules (skip connections), which makes the network prone to spectral distortion on new datasets. Whereas, after the introduction of a skip connection (A-PNN) and conducting a fine-tuning strategy, the network (*i.e.,* the A-PNN-FT) can get better results (reaching the fourth position in the ranking), thus corroborating the generalization ability of the adaptive fine-tuning combined with the robustness provided by properly set residual skip connections.

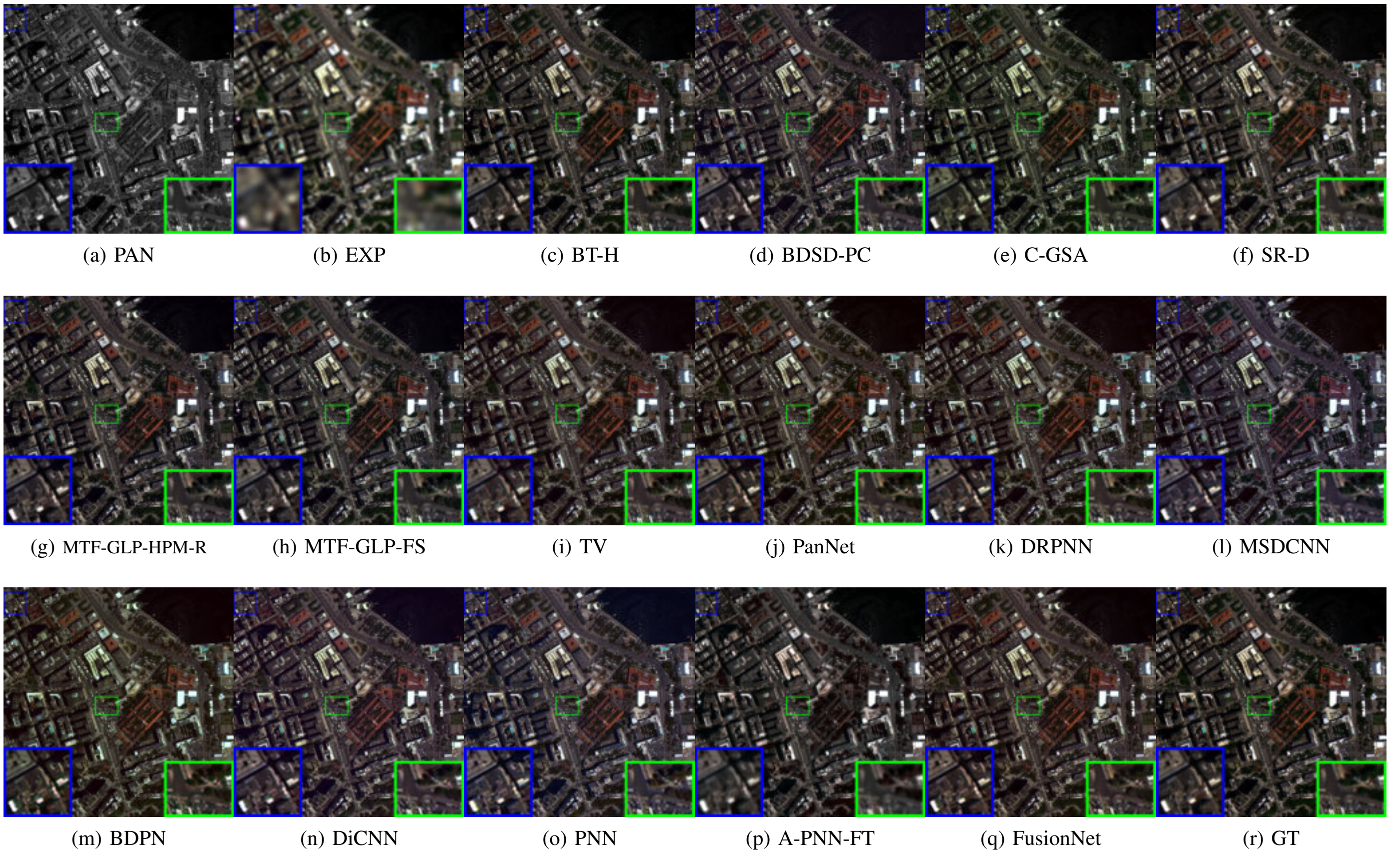
TABLE V: Quantitative comparison of the outcomes of the benchmark on the reduced resolution WV2 Rio dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q8	SAM	ERGAS
CS/MRA/VO			
GT	1.0000	0.0000	0.0000
EXP	0.7283	4.8597	6.7878
BT-H	**0.9441**	3.5368	3.3027
BDSD-PC	0.9316	4.0320	3.7105
C-GSA	0.9407	3.8848	3.3972
SR-D	0.9375	3.7881	3.3127
MTF-GLP-HPM-R	0.9436	3.8778	**3.2173**
MTF-GLP-FS	0.9426	3.8129	3.2578
TV	0.9341	4.1811	3.7521
ML			
PanNet	0.9329	4.2582	3.8532
DRPNN	0.9301	5.0920	4.1910
MSDCNN	0.9200	5.4779	3.8565
BDPN	0.8888	5.9709	5.5306
DiCNN	0.8925	5.6765	5.4202
PNN	0.8866	9.4634	6.5718
A-PNN-FT	0.9374	**3.5300**	3.3032
FusionNet	0.9069	5.1220	4.1184

G. Assessment on WV3 Data

In this section, we repeat the same three tests as in Sect. IV-F, but involving WV3 data. Multiple reduced resolution testing datasets are evaluated first. Then, another dataset is used to assess the performance both at reduced resolution and at full resolution.

1) Performance on 4 Reduced Resolution Testing Datasets: This section will evaluate first all the compared pansharpening methods on 4 reduced resolution WV3 Rio testing datasets sharing a similar geographic area and the same acquiring time as that of one of the datasets used for the training (see the testing data in Tab. II (B) and the WV3 Rio image in Fig.


(a) PAN (b) EXP (c) BT-H (d) BDSD-PC (e) C-GSA (f) SR-D

(g) MTF-GLP-HPM-R (h) MTF-GLP-FS (i) TV (j) PanNet (k) DRPNN (l) MSDCNN

(m) BDPN (n) DiCNN (o) PNN (p) A-PNN-FT (q) FusionNet (r) GT

Fig. 13: Visual comparisons in natural colors of the compared approaches on the reduced resolution WV2 Rio dataset, see also Fig. 11.

10). Tab. VII reports the average numerical results of all the sixteen compared state-of-the-art pansharpening methods on the 4 reduced resolution WV3 Rio testing datasets. From the table, it is clear that all the ML-based approaches outperform traditional methods on all the related three reduced resolution indicators, *i.e.,* Q8, SAM and ERGAS. Note that FusionNet gets the best Q8, SAM, and ERGAS metrics (and also most of the best stds), showing its promising ability on the testing data that are acquired over similar geographic areas as those of the ones represented in the training data. Among all the ML-based methods, PanNet, DRPNN, MSDCNN, and DiCNN can be grouped in a second-best class. Indeed, the performance is slightly worse than FusionNet and A-PNN-FT but better than BDPN and PNN. Among traditional methods, BT-H outperforms the others. Besides, TV gets the worst Q8 and ERGAS indicators. The main reason of the outstanding performance of ML-based methods is the same as in Sect. IV-F1, *i.e.,* the similarity between training and testing datasets.

2) Performance on the Reduced Resolution WV3 New York Dataset: We evaluate the quantitative performance of all the compared pansharpening methods on a new reduced resolution WV3 dataset used only for testing purposes acquired over the city of New York and shown in Fig. 11. The testing dataset shows a different geographical area captured in a different time comparing with the training dataset. By looking at Tab. VIII, the traditional approaches outperform most of the ML-based methods on all the quality metrics. BDSD-PC, belonging to the class of traditional methods, gets two best indicators, *i.e.,* Q8 and ERGAS, while another traditional technique, *i.e.,* the BT-H, gets the lowest SAM. Nevertheless, some ML-based approaches, such as PanNet, DRPNN, BDPN, and A-PNN-FT, also obtain competitive outcomes representing the second-best class among all the compared methods. In contrast, the other three ML-based methods, such as, DiCNN, PNN, and FusionNet, get the worst performance. Especially, PNN shows the largest SAM (almost 5 degrees more than the second-worst method), indicating a higher spectral distortion than the other methods. The reason why the mentioned three ML-based methods are worse than the other ML-based methods is because their simpler network architecture with less parameters, which could not fit well the problem's non-linearities.

3) Performance on the Full Resolution WV3 New York Dataset: Similar to Sect. IV-F3, this section will compare the qualitative and quantitative performance of all the methods on the full resolution WV3 New York dataset, see Fig. 11. This full resolution testing dataset is acquired over the city of New York showing a different geographical area and acquisition time comparing them with the ones of the training data. Again, the HQNR is used for performance assessment. Tab. IX shows that the traditional techniques outperform most of

TABLE VI: Quantitative comparison of the outcomes of the benchmark on the full resolution WV2 Rio dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	D_λ	D_S	**HQNR**
CS/MRA/VO			
EXP	0.0374	0.0717	0.8936
BT-H	0.0601	0.0710	0.8732
BDSD-PC	0.0653	0.0435	0.8940
C-GSA	0.0664	0.0653	0.8727
SR-D	**0.0153**	0.0286	**0.9566**
MTF-GLP-HPM-R	0.0260	0.0594	0.9161
MTF-GLP-FS	0.0269	0.0652	0.9097
TV	0.0332	0.0269	0.9407
ML			
PanNet	<u>0.0292</u>	**<u>0.0171</u>**	<u>0.9542</u>
DRPNN	0.0629	0.0311	0.9080
MSDCNN	0.0872	0.0498	0.8674
BDPN	0.0909	0.0486	0.8649
DiCNN	0.1043	0.0478	0.8529
PNN	0.1678	0.0510	0.7897
A-PNN-FT	0.0379	0.0396	0.9240
FusionNet	0.0647	0.0179	0.9185

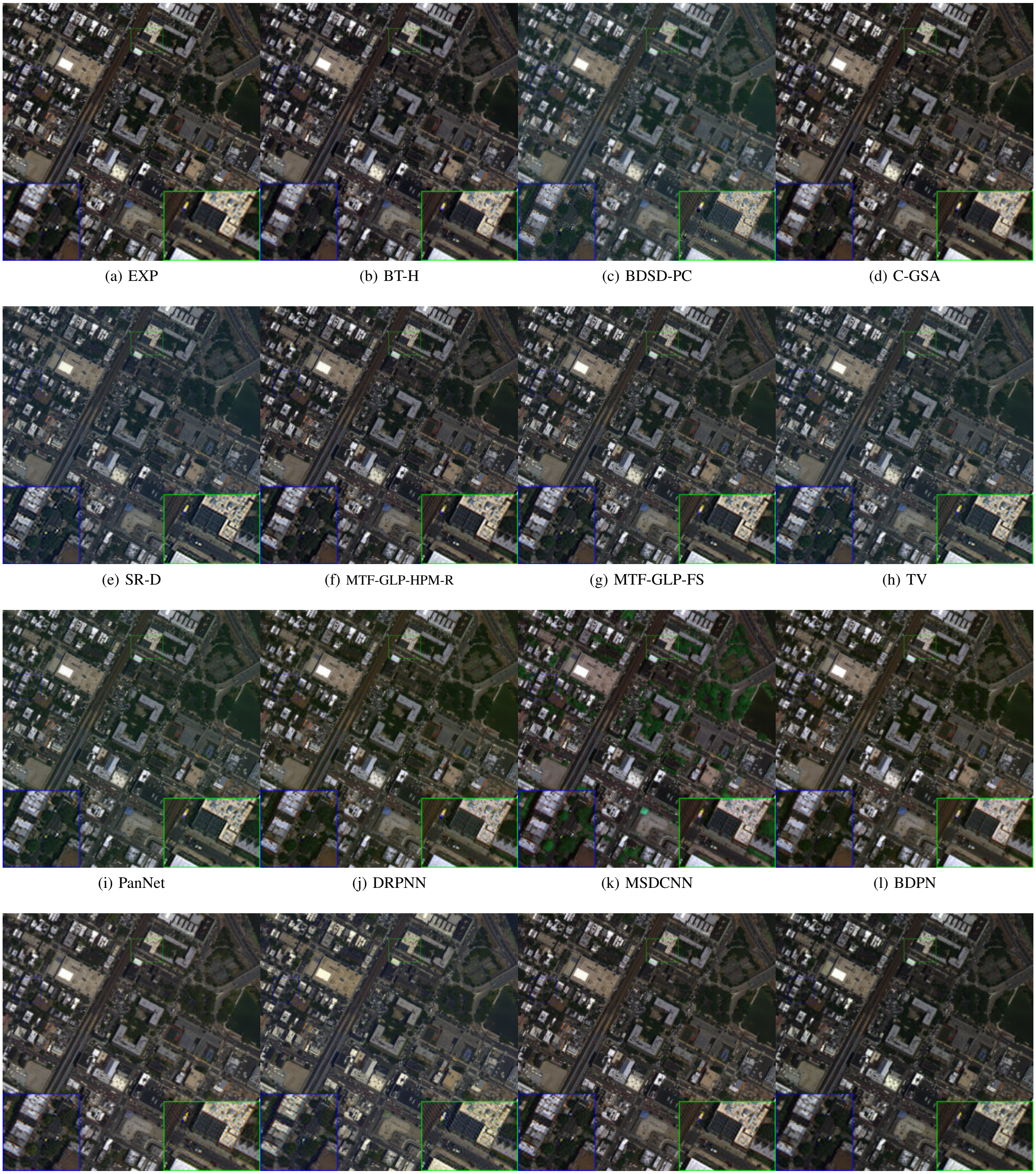
TABLE VII: Average results for the approaches belonging to the benchmark on the reduced resolution WV3 Rio testing dataset, *i.e.*, on the 4 WV3 testing datasets in Tab. II (B). Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q8 (± std)	**SAM** (± std)	**ERGAS** (± std)
CS/MRA/VO			
GT	1.0000±0.0000	0.0000±0.0000	0.0000±0.0000
EXP	0.5974±0.0571	9.2031±0.7655	9.3369±0.4756
BT-H	0.8898±0.0323	7.6700±0.7613	4.6132±0.1695
BDSD-PC	0.8454±0.0608	8.9376±0.8568	5.0893±0.2015
C-GSA	0.8695±0.0436	8.8042±0.8652	5.0183±0.1327
SR-D	0.8693±0.0320	7.9449±0.4946	5.0739±0.2807
MTF-GLP-HPM-R	0.8625±0.0499	9.4911±1.1386	5.2141±0.2881
MTF-GLP-FS	0.8533±0.0526	9.1442±0.9443	5.2496±0.2077
TV	0.8031±0.0929	8.9863±0.8592	5.3569±0.1948
ML			
PanNet	0.9232±0.0324	5.1447±0.3995	3.1906±0.2192
DRPNN	0.9203±0.0323	5.1492±0.3500	3.2171±0.2067
MSDCNN	0.9154±0.0351	5.5887±0.3471	3.3978±0.1715
BDPN	0.9137±0.0327	6.0121±0.4713	3.6072±0.2425
DiCNN	0.9265±0.0283	5.1285±0.3217	3.1894±0.2106
PNN	0.9068±0.0425	5.9259±0.4544	3.4998±0.1341
A-PNN-FT	**<u>0.9327±0.0255</u>**	4.9125±0.3794	3.0880±0.2312
FusionNet	**<u>0.9327±0.0272</u>**	**<u>4.6482±0.3508</u>**	**<u>2.9028±0.1967</u>**

TABLE VIII: Quantitative comparison of the outcomes of the benchmark on the reduced resolution WV3 New York dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q8	**SAM**	**ERGAS**
CS/MRA/VO			
GT	1.0000	0.0000	0.0000
EXP	0.6513	7.2118	8.1106
BT-H	0.9241	**6.4530**	3.9714
BDSD-PC	**0.9327**	6.8388	**3.8905**
C-GSA	0.9213	6.6966	4.0503
SR-D	0.9113	6.6269	4.3472
MTF-GLP-HPM-R	0.9228	7.0038	4.0692
MTF-GLP-FS	0.9228	6.7650	4.0434
TV	0.9277	6.6213	4.0630
ML			
PanNet	<u>0.9238</u>	<u>6.9050</u>	<u>4.2365</u>
DRPNN	0.9205	7.3887	4.2504
MSDCNN	0.9087	7.5139	4.4214
BDPN	0.9180	7.7148	4.4522
DiCNN	0.8567	8.0256	5.5124
PNN	0.8849	12.6019	6.7233
A-PNN-FT	0.9132	7.6201	4.4536
FusionNet	0.8499	8.3823	6.0458

the ML-based methods. The best results are reported by the TV. Instead, the SR-D got the third position in the ranking. Considering all the methods, the traditional MTF-GLP-HPM-R and MTF-GLP-FS techniques also show superior performance than the ML-based approaches except for the PanNet and the A-PNN-FT, which obtained the second and the fourth positions, respectively. Among all the ML-based techniques, the PanNet and the A-PNN-FT got the first two positions thanks to their competitive generalization abilities. Moreover, some ML-based methods, including MSDCNN, DiCNN, and PNN, have relatively large gaps comparing them with the PanNet and the A-PNN-FT methods. The rest of the ML-based methods, such as DRPNN, BDPN, and FusionNet, achieve a similar performance than the traditional methods as BT-H, BDSD-PC, and C-GSA . Additionally, Fig. 14 displays the visual comparison of all the compared methods on the full resolution WV3 New York testing dataset. By having a look at this figure, the traditional TV method getting the best HQNR has not the most clear fused image comparing it with the ML-based ones, see the blue and green close-ups. BDSD-PC seems to have a blur effect and a clear spectral distortion (mainly due to a color contrast changing). A relevant spectral distortion also happens in the case of the pansharpened SR-D product. Furthermore, BH-T seems to get precise spatial details even though its quantitative outcomes are not so promising. The visual products of the ML-based techniques are quite competitive without showing a significant spectral distortion. However, some methods, such as, the DiCNN and the PNN, generate significant blur effects and artifacts (like outliers) indicating a weak visual appearance. Finally, BDPN shows the clearest spatial details without any artifact. Note that the rest of the ML-based approaches practically yield similar visual performance, showing clear spatial details and a good spectral preservation.



(a) EXP (b) BT-H (c) BDSD-PC (d) C-GSA

(e) SR-D (f) MTF-GLP-HPM-R (g) MTF-GLP-FS (h) TV

(i) PanNet (j) DRPNN (k) MSDCNN (l) BDPN

(m) DiCNN (n) PNN (o) A-PNN-FT (p) FusionNet

Fig. 14: Visual comparisons in natural colors of the compared approaches on the full resolution WV3 New York dataset, see also Fig. 11.

H. Assessment on WV4 Data

In this section, we repeat the same three tests as in Sect. IV-F, but involving WV4 data. Multiple reduced resolution testing datasets are evaluated first. Then, another dataset is used to assess the performance both at reduced resolution and

TABLE IX: Quantitative comparison of the outcomes of the benchmark on the full resolution WV3 New York dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	D_λ	D_S	**HQNR**
CS/MRA/VO			
EXP	0.0562	0.1561	0.7964
BT-H	0.0983	0.0829	0.8269
BDSD-PC	0.1554	0.0251	0.8234
C-GSA	0.1022	0.0747	0.8307
SR-D	**0.0199**	0.0369	0.9440
MTF-GLP-HPM-R	0.0356	0.0679	0.8989
MTF-GLP-FS	0.0347	0.0740	0.8939
TV	0.0234	0.0252	**0.9520**
ML			
PanNet	0.0376	**0.0162**	0.9468
DRPNN	0.1207	0.0392	0.8449
MSDCNN	0.1583	0.0557	0.7948
BDPN	0.1338	0.0563	0.8175
DiCNN	0.1023	0.0979	0.8098
PNN	0.1465	0.0835	0.7823
A-PNN-FT	0.0510	0.0198	0.9302
FusionNet	0.0941	0.0882	0.8260

TABLE X: Average results for the approaches belonging to the benchmark on the reduced resolution WV4 Acapulco testing dataset, *i.e.*, on the 8 WV4 testing datasets in Tab. II (C). Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q4 (± std)	**SAM** (± std)	**ERGAS** (± std)
CS/MRA/VO			
GT	1.0000±0.0000	0.0000±0.0000	0.0000±0.0000
EXP	0.2638±0.1579	3.9822±0.5496	4.7990±0.9197
BT-H	0.6499±0.0734	4.3582±0.5607	4.6330±0.8555
BDSD-PC	0.6512±0.0680	3.6968±0.5468	4.1906±0.9251
C-GSA	0.6528±0.0638	3.7530±0.5137	4.4599±0.8186
SR-D	0.6564±0.0895	3.6514±0.4579	3.9887±0.7342
MTF-GLP-HPM-R	0.6698±0.0606	3.7980±0.6864	4.2282±0.8625
MTF-GLP-FS	0.6666±0.0598	3.7776±0.6527	4.2159±0.8816
TV	0.5125±0.1459	4.0344±0.5386	4.2600±0.6768
ML			
PanNet	**0.6963±0.0842**	**3.3710±0.4221**	**3.6088±0.6313**
DRPNN	0.6810±0.0845	3.4778±0.4499	3.6706±0.6433
MSDCNN	0.6739±0.0849	3.4837±0.4601	3.7052±0.6661
BDPN	0.6535±0.0834	3.5222±0.4612	3.8269±0.7144
DiCNN	0.6767±0.0832	3.4555±0.4453	3.7087±0.6629
PNN	0.6793±0.0822	3.4777±0.4589	3.6894±0.6613
A-PNN-FT	0.6787±0.0820	3.4271±0.4425	3.6995±0.6705
FusionNet	0.6759±0.0805	3.3979±0.4442	3.6842±0.6760

at full resolution.

1) Performance on 8 Reduced Resolution Testing Datasets: After evaluating the performance of the 8-band WV2 and WV3 datasets, this section mainly focuses on comparing the performance of the 4-band WV4 dataset acquired over the city of Acapulco, Mexico. Although we have a different spectral band number with respect to the 8-band datasets in Sects. IV-F and IV-G, the whole testing procedure follows the same direction as in the previous two sections. Indeed, all the compared pansharpening methods will be evaluated on 8 reduced resolution samples extracted from the WV4 Acapulco testing dataset shown in Tab. II (C) and Fig.10. These testing datasets share similar features with the training data, again. Tab. X reports the quantitative comparison showing that the compared ML-based approaches yield better performance than that of the traditional techniques. PanNet, belonging to the ML class, gets the best indicators among all the methods. Besides, the rest of ML-based methods, *i.e.,* DRPNN, MSDCNN, BDPN, DiCNN, PNN, A-PNN-FT, and FusionNet, get similar performance, yielding small gaps among the three metrics exploited at reduced resolution. DRPNN obtained the second-best Q4 and ERGAS indicators. Instead, FusionNet got the second-best SAM. Among all the traditional approaches, MTF-GLP-HPM-R obtained the best Q4 and SR-D got the best SAM and ERGAS. Moreover, the same conclusion as in Sect. IV-F1 about the relationship between the performance of ML-based approaches and traditional methods can be drawn.

2) Performance on the Reduced Resolution WV4 Alice Springs Dataset: This section still investigates on the performance of all the methods on a different reduced resolution WV4 dataset acquired over the city of Alice Springs, another area of the world with respect to the training dataset. Readers can refer to Fig. 11. By having a look at Tab. XI, the ML-based methods, *i.e.,* DRPNN and PanNet, get the best Q4 and SAM, respectively, while the traditional SR-D method has the best ERGAS. Overall, the quantitative performance of all the methods is similar among each other. No approach obtains the best outcomes on all the indexes. For example, some ML-based methods, *e.g.,* A-PNN-FT, PanNet, and PNN, get better SAM than several traditional methods, *e.g.,* BDSD-PC, C-GSA, and MTF-GLP-FS, whereas some traditional methods, *e.g.,* BT-H and SR-D, obtain better SAM than some ML-based methods, such as, DRPNN, DiCNN, and FusionNet. Among the ML-based methods, although DRPNN achieves the best Q4, its SAM value is significantly lower than those of the PanNet and the A-PNN-FT. Besides, FusionNet yields the worst metrics among all the ML-based methods. Fig. 15 shows the visual comparison of all the pansharpening approaches showing that all the methods obtain excellent results with high spatial fidelity on the urban area. In particular, traditional methods, such as, BT-H, C-GSA, MTF-GLP-HPM-R, and MTF-GLP-FS display products with clearer spatial details than the ML-based methods, see the close-ups in Fig. 15. Moreover, some other traditional methods, such as, SR-D and TV, show a significant blur, see the blur and green close-ups in Fig. 15.

3) Performance on the Full Resolution WV4 Alice Springs Dataset: Tab. XII reports the quantitative results on the WV4 Alice Spring dataset using data at the original (full) resolution, see Fig. 11. Note that due to the absence of GT image, we employ the no reference indicators, such as, the HQNR, the D_λ, and the D_S to evaluate the quantitative performance.

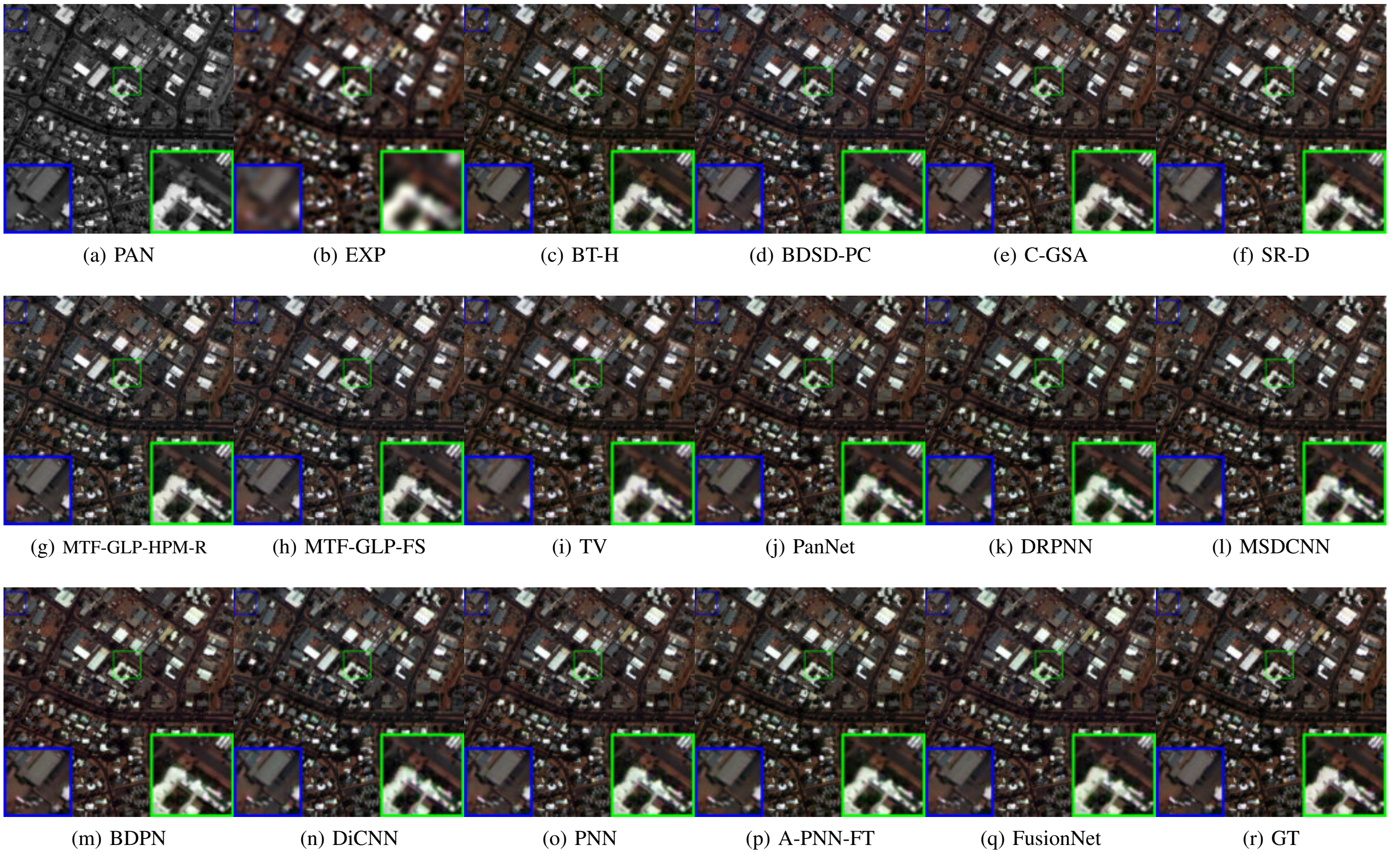


Fig. 15: Visual comparisons in natural colors of the compared approaches on the reduced resolution WV4 Alice Springs dataset, see also Fig. 11.

From the table, it is clear that some traditional and ML-based methods, such as SR-D, TV, and A-PNN-FT, achieve the highest values of the HQNR index. Moreover, most of the ML-based approaches get better indexes than the rest of the traditional techniques, *i.e.,* BT-H, BDSD-PC, and C-GSA. Among all the ML-based methods, the A-PNN-FT, the PanNet, and the DRPNN belong to the best performance class. Moreover, the MSDCNN, the BDPN, and the DiCNN represent the second-best class, while the rest of the ML-based approaches (*i.e.,* the PNN and the FusionNet) get the lowest performance. Finally, the A-PNN-FT obtains the best ML-based quantitative outcome corroborating the effectiveness of the use of the fine-tuning strategy.

I. Assessment on QB Data

This section investigates first the performance on both reduced resolution and full resolution testing sets, similarly as the analysis conducted before. Then, we also evaluate the ability of the compared networks to generalize with respect to the acquisition sensor. Indeed, we will exploit ML-based methods trained on the QB training set, but evaluating them on another 4-band dataset acquired by the IKONOS sensor.

1) Performance on 7 Reduced Resolution Testing Datasets: This section focuses on the testing on 7 reduced resolution QB Indianapolis datasets that can be found in Fig. 10. Again, these testing datasets have a similar area and the same acquisition time as that of the training dataset (see the data ① in Tab. II). Due to this reason, the outcomes of the ML-based approaches get better quantitative results than that of the compared traditional methods, see Tab. XIII. FusionNet gets the best Q4, SAM, and ERGAS indicators, and PanNet, DRPNN, MSDCNN, DiCNN, and A-PNN-FT represent the second-best class. Besides, comparing the mentioned ML-based approaches, BDPN gets relatively lower performance than the other ML-based methods, but still outperforming the traditional techniques.

2) Performance on the Reduced Resolution QB San Francisco Dataset: These results are about the assessment of all the methods on another reduced resolution dataset acquired by the QB sensor over the city of San Francisco (USA), see Fig. 11. In Tab. XIV, we can note that the traditional approaches have better quantitative results than those of the ML-based methods (except for the PanNet). Besides, C-GSA and BT-H methods get the lowest and the second-lowest ERGAS, respectively. Among the ML-based methods, PanNet has the best Q4, SAM, and ERGAS indicators, even better than those of all the traditional approaches. For the other ML-based methods, none generates the best outcomes on all the indexes. The quantitative results for the rest of the ML-based approaches are not stable. For instance, DRPNN yields the second-best

TABLE XI: Quantitative comparison of the outcomes of the benchmark on the reduced resolution WV4 Alice Springs dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q4	**SAM**	**ERGAS**
CS/MRA/VO			
GT	1.0000	0.0000	0.0000
EXP	0.7901	4.5880	5.8077
BT-H	0.9444	4.1988	3.2757
BDSD-PC	0.9431	4.7527	3.2996
C-GSA	0.9417	4.8812	3.3223
SR-D	0.9493	4.1597	**3.0647**
MTF-GLP-HPM-R	0.9432	5.1721	3.2724
MTF-GLP-FS	0.9432	4.9296	3.2437
TV	0.9250	4.7857	3.6899
ML			
PanNet	0.9486	**3.8737**	3.4154
DRPNN	**0.9521**	4.7059	3.2533
MSDCNN	0.9266	4.5170	3.9181
BDPN	0.9439	4.4687	3.5570
DiCNN	0.9347	4.8219	3.6978
PNN	0.9364	4.4324	3.5983
A-PNN-FT	0.9511	3.9217	3.1598
FusionNet	0.9261	4.9779	3.9561

TABLE XII: Quantitative comparison of the outcomes of the benchmark on the full resolution WV4 Alice Springs dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	D_λ	D_S	**HQNR**
CS/MRA/VO			
EXP	0.0362	0.0322	0.9328
BT-H	0.0585	0.0625	0.8826
BDSD-PC	0.0644	0.0435	0.8950
C-GSA	0.0668	0.0796	0.8589
SR-D	**0.0109**	0.0331	**0.9564**
MTF-GLP-HPM-R	0.0229	0.0608	0.9177
MTF-GLP-FS	0.0230	0.0623	0.9161
TV	0.0251	**0.0237**	0.9518
ML			
PanNet	0.0120	0.0429	0.9456
DRPNN	0.0223	0.0333	0.9452
MSDCNN	0.0221	0.0641	0.9152
BDPN	0.0260	0.0498	0.9255
DiCNN	0.0455	0.0358	0.9203
PNN	0.0195	0.0722	0.9097
A-PNN-FT	0.0195	0.0306	0.9505
FusionNet	0.0668	0.0274	0.9076

Q4 among all the ML-based methods, but its SAM value is clearly larger than that of the FusionNet.

TABLE XIII: Average results for the approaches belonging to the benchmark on the reduced resolution QB Indianapolis testing dataset, *i.e.*, on the 7 QB testing datasets in Tab. II (D). Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q4 (± std)	**SAM** (± std)	**ERGAS** (± std)
CS/MRA/VO			
GT	1.0000±0.0000	0.0000±0.0000	0.0000±0.0000
EXP	0.7490±0.0170	4.5865±0.4136	4.0991±0.1660
BT-H	0.8729±0.0102	3.7376±0.4094	3.0172±0.1599
BDSD-PC	0.8643±0.0107	4.0724±0.5258	3.2261±0.1212
C-GSA	0.8307±0.0367	4.4207±0.7011	3.5343±0.4436
SR-D	0.8789±0.0091	3.6989±0.3694	2.9774±0.1687
MTF-GLP-HPM-R	0.8628±0.0151	3.9175±0.7783	3.2746±0.4262
MTF-GLP-FS	0.8513±0.0152	4.0604±0.7747	3.3176±0.1050
TV	0.8049±0.0371	4.8419±0.3162	3.9387±0.4611
ML			
PanNet	0.9575±0.0072	1.9853±0.1919	1.7365±0.0880
DRPNN	0.9510±0.0086	2.0873±0.1875	1.8378±0.0916
MSDCNN	0.9509±0.0088	2.0771±0.1810	1.8565±0.1025
BDPN	0.9238±0.0113	2.5859±0.1981	2.3305±0.1474
DiCNN	0.9510±0.0088	2.0704±0.1793	1.8764±0.1086
PNN	0.9487±0.0085	2.1556±0.1850	1.9054±0.1048
A-PNN-FT	0.9585±0.0074	1.8825±0.1676	1.7086±0.0963
FusionNet	**0.9600±0.0082**	**1.8298±0.1391**	**1.6470±0.0918**

TABLE XIV: Quantitative comparison of the outcomes of the benchmark on the reduced resolution QB San Francisco dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q4	**SAM**	**ERGAS**
CS/MRA/VO			
GT	1.0000	0.0000	0.0000
EXP	0.5759	9.1351	10.9039
BT-H	0.8942	7.5545	5.2697
BDSD-PC	0.8788	8.7450	5.7536
C-GSA	0.8961	7.4711	**5.2337**
SR-D	0.8831	7.8766	5.5558
MTF-GLP-HPM-R	0.8919	8.4890	5.4754
MTF-GLP-FS	0.8770	8.7026	5.7855
TV	0.8802	8.4317	6.0476
ML			
PanNet	**0.9074**	**6.9841**	5.3314
DRPNN	0.8969	8.2530	5.9467
MSDCNN	0.8768	7.5988	5.6965
BDPN	0.8830	8.4378	5.9962
DiCNN	0.8062	11.2110	8.7013
PNN	0.8301	10.1118	6.8375
A-PNN-FT	0.8586	7.8767	6.2049
FusionNet	0.8614	7.3459	6.3420

3) Performance on the Full Resolution QB San Francisco Dataset: The QB San Francisco dataset shown in Fig. 11 is also used at full resolution. From Tab. XV, reporting all the no reference indexes, it is easy to see that the PanNet method obtains the best no reference index, *i.e.,* the HQNR, which means the best quantitative outcome. Moreover, the

TABLE XV: Quantitative comparison of the outcomes of the benchmark on the full resolution QB San Francisco dataset, see also Fig. 11. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	D_λ	D_S	**HQNR**
CS/MRA/VO			
EXP	0.0470	0.1571	0.8033
BT-H	0.0925	0.0925	0.8236
BDSD-PC	0.1383	0.0476	0.8207
C-GSA	0.0818	0.1114	0.8159
SR-D	**0.0144**	0.0362	0.9499
MTF-GLP-HPM-R	0.0343	0.1126	0.8570
MTF-GLP-FS	0.0372	0.1323	0.8354
TV	0.0269	0.0513	0.9233
ML			
PanNet	0.0224	0.0264	**0.9518**
DRPNN	0.0662	0.0210	0.9142
MSDCNN	0.0771	0.0268	0.8982
BDPN	0.0621	0.0708	0.8715
DiCNN	0.0939	0.1244	0.7933
PNN	0.1071	0.0671	0.8330
A-PNN-FT	0.0364	0.0303	0.9344
FusionNet	0.0388	**0.0139**	0.9479

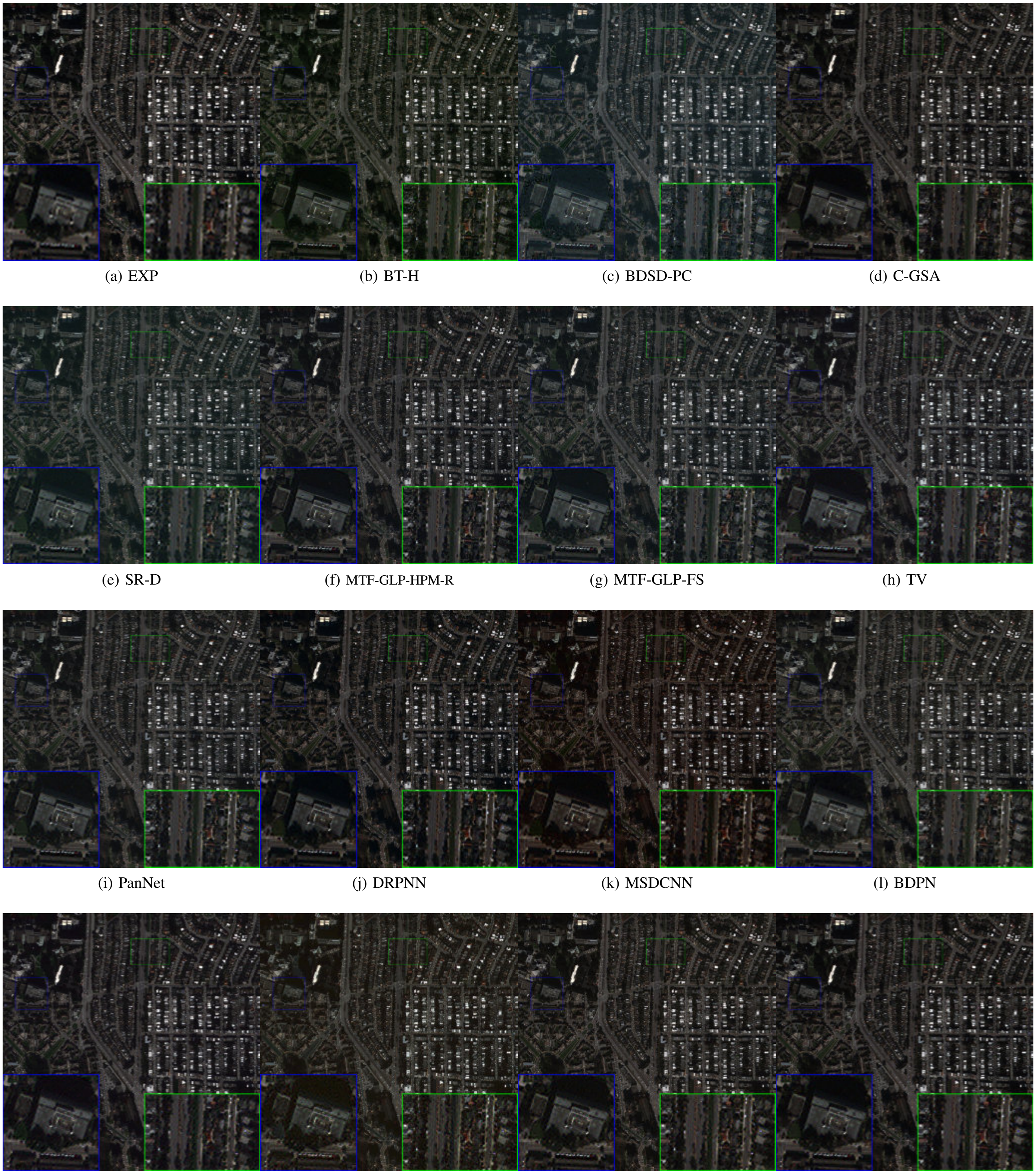
TABLE XVI: Quantitative comparison of the outcomes of the benchmark on the reduced resolution IKONOS Toulouse dataset, see also Fig. 11. The ML-based approaches are trained on the QB dataset. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	Q4	**SAM**	**ERGAS**
CS/MRA/VO			
GT	1.0000	0.0000	0.0000
EXP	0.4795	5.1823	6.3953
BT-H	**0.9120**	3.4491	2.9962
BDSD-PC	0.9094	2.9576	2.9309
C-GSA	0.9006	2.9667	3.1751
SR-D	0.9108	2.9571	**2.8708**
MTF-GLP-HPM-R	0.9105	3.1454	2.9727
MTF-GLP-FS	0.9076	3.0906	3.0104
TV	0.9023	**2.8455**	2.9508
ML			
PanNet	0.8826	3.9010	3.6584
DRPNN	0.8884	5.9745	4.2175
MSDCNN	0.8736	4.1837	3.5057
BDPN	0.8783	4.0874	3.7266
DiCNN	0.8143	6.2024	5.5863
PNN	0.8406	4.6105	3.9881
A-PNN-FT	0.8838	3.6224	3.3742
FusionNet	0.8159	4.2536	4.0710

traditional method SR-D and the ML-based method FusionNet rank the second and the third places, respectively. Overall, the traditional methods (except for the SR-D and the TV) obtain lower performance than most of the ML-based methods. The HQNR got by the DiCNN method is the lowest one demonstrating that the learned weights of the DiCNN network cannot fit the problem presented during the testing phase. In addition, Fig. 16 exhibits the visual comparison of all the compared methods on the full resolution QB San Francisco dataset. From the figure, the ML-based methods, *i.e.,* PanNet and FusionNet, retain the clearest details, consistently with the HQNR performance in Tab. XV. Moreover, most of the other ML-based methods, such as, DRPNN, MSDCNN, BDPN, and A-PNN-FT, preserve the spatial content. Only DiCNN and PNN seem to have relatively noticeable blur effects and artifacts. Among the traditional methods, BDSD-PC shows a significant spectral distortion. Instead, the two MTF-based techniques, *i.e.,* MTF-GLP-HPM-R and MTF-GLP-FS, get a high spatial fidelity, although they fail to get promising HQNR values. Finally, the TV and the SR-D get a similar spatial preservation as that of the A-PNN-FT.

4) Sensor Generalization Ability Assessed on the Reduced Resolution IKONOS Dataset: This section evaluates the network generalization ability for all the compared ML-based methods. The latter are trained on the 4-band QB training set used in the above-mentioned sections. Then, we directly test the ML-based approaches running them on a 4-band IKONOS dataset acquired over the city of Toulouse, France. Besides, we also compare the ML-based methods with some traditional techniques. From Tab. XVI, it is clear that BT-H, TV, and SR-D get the best Q4, SAM, and ERGAS, respectively. In contrast, ML-based methods have quite low performance demonstrating a weak network generalization. Overall, traditional methods outperform all the ML-based approaches. Among the ML-based techniques, PanNet and A-PNN-FT yield the best quantitative results on the three quality metrics. The other ML-based methods obtain lower performance. Fig. 17 depicts the fused products showing competitive performance for some traditional methods, *i.e.,* the BT-H, the C-GSA, the MTF-GLP-HPM-R, and the MTF-GLP-FS. Although the SR-D has the best ERGAS, some artifacts appear in the related outcome (see the blue close-up in Fig. 17). Among the ML-based methods, all the compared approaches have similar spatial details. However, some of them, such as, DRPNN, DiCNN, and FusionNet, have a significant spectral distortion (see the color of the river in Fig. 17). This is also corroborated by the SAM values in Tab. XVI.

5) Sensor Generalization Ability Assessed on the Full Resolution IKONOS Dataset: The same analysis as in Sect. IV-I4 is performed at full resolution exploiting the IKONOS Toulouse dataset, again. The A-PNN-FT yields the best overall performance, see Tab. XVII. Indeed, thanks to the use of the fine-tuning strategy, A-PNN-FT has a better network generalization ability than the other ML techniques. This is a good hint for future developments that could include this strategy to increase the generalization ability. Another ML approach getting competitive performance with respect to traditional methods is the PanNet. About the traditional methods, the SR-D obtains the highest performance. Moreover, other two traditional approaches, *i.e.,* C-GSA and TV, also achieve promising results. Finally, it is worth to be pointed out



(a) EXP (b) BT-H (c) BDSD-PC (d) C-GSA

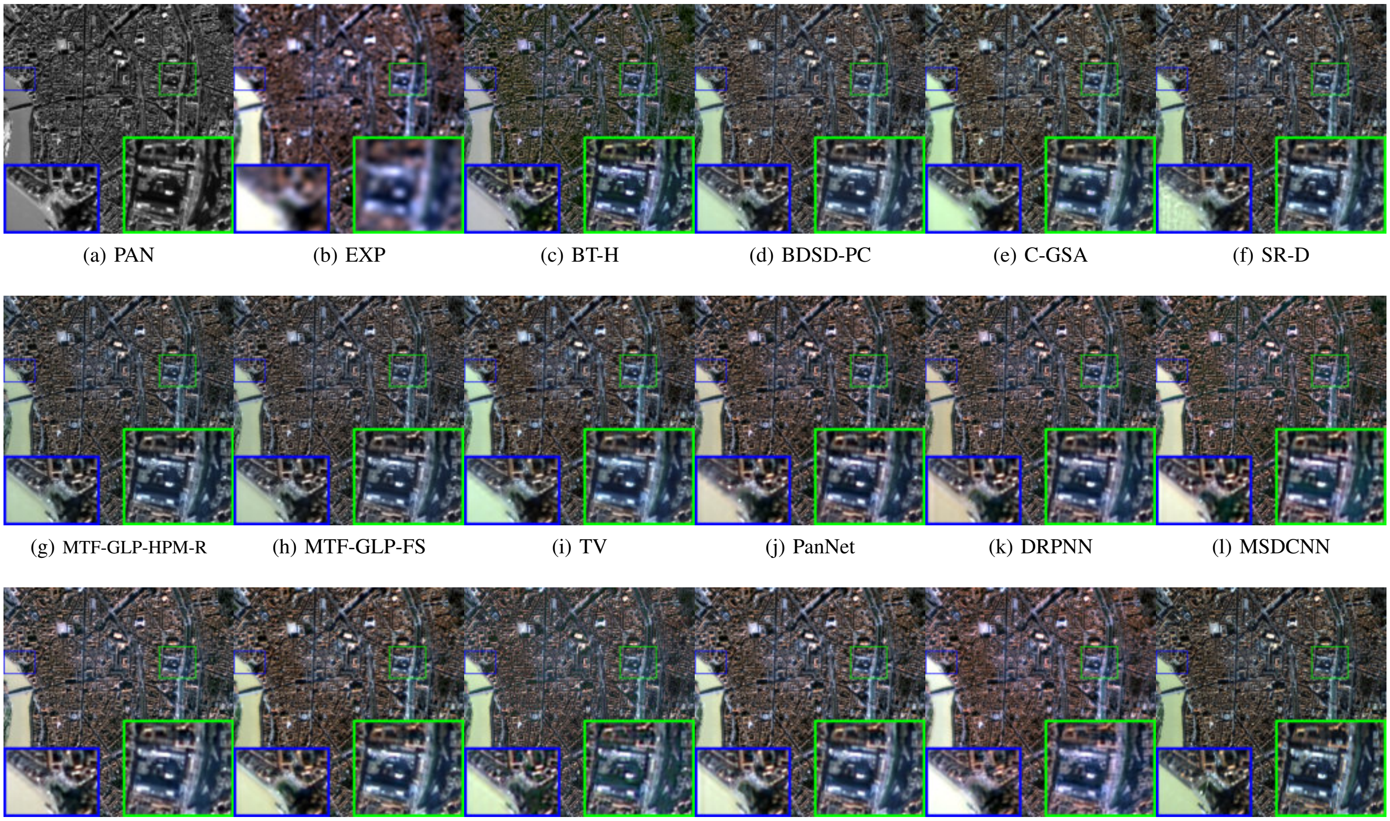
(e) SR-D (f) MTF-GLP-HPM-R (g) MTF-GLP-FS (h) TV

(i) PanNet (j) DRPNN (k) MSDCNN (l) BDPN

(m) DiCNN (n) PNN (o) A-PNN-FT (p) FusionNet

Fig. 16: Visual comparisons in natural colors of the compared approaches on the full resolution San Francisco dataset, see also Fig. 11.

that despite of the HQNR index represents a state-of-the-art quality index, more research is still needed about this topic [4]. Indeed, the difficult in ranking approaches belonging to very different philosophies (*e.g.*, classical against ML methods) is



(a) PAN (b) EXP (c) BT-H (d) BDSD-PC (e) C-GSA (f) SR-D

(g) MTF-GLP-HPM-R (h) MTF-GLP-FS (i) TV (j) PanNet (k) DRPNN (l) MSDCNN

(m) BDPN (n) DiCNN (o) PNN (p) A-PNN-FT (q) FusionNet (r) GT

Fig. 17: Visual comparisons in natural colors of the compared approaches on the reduced resolution IKONOS Toulouse dataset, see also Fig. 11.

evident. Thus, results at reduced and full resolution can hardly be compared when referring to methods in different classes.

J. Discussions

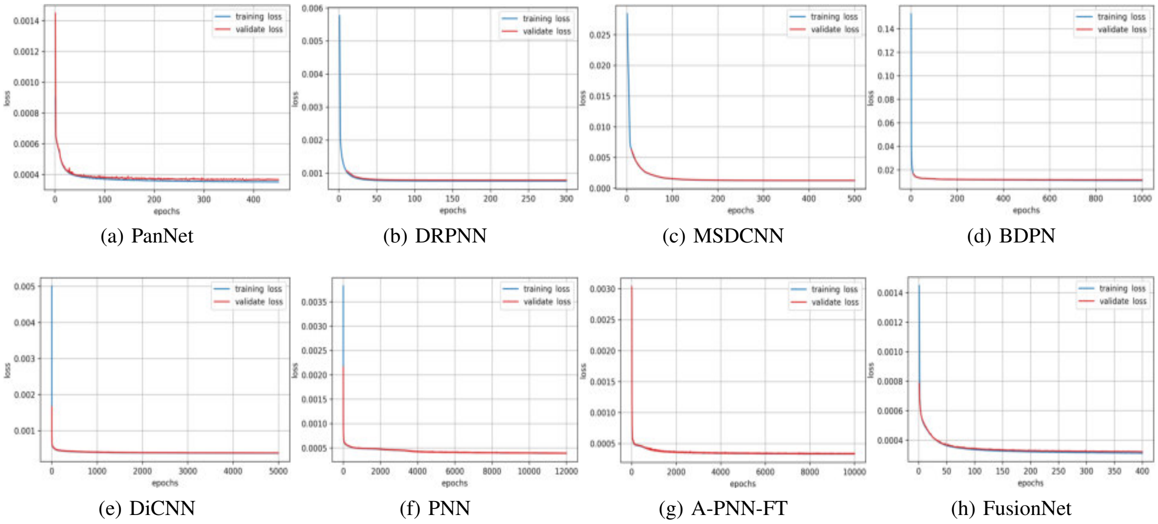
This section is devoted to some final discussions about the ML-based approaches. Some aspects as convergence, testing and training times, amount of parameters, and so forth will be detailed in the following.

1) Convergence: Fig. 18 exhibits the training loss and validation loss of all the compared ML-based approaches. The goal of this analysis is to show that the ML approaches converge but avoiding the overfitting phenomenon. Observing the curves depicted in Fig. 18, we can state that the goal is achieved by all the compared methods.

2) Testing Time: To evaluate the testing time of all the compared pansharpening methods, we employ 4 reduced resolution WV3 testing datasets, see Sect. IV-G1 for more details. Tab. XVIII reports the average testing time for all the compared methods. Note that the traditional approaches are implemented on the CPU, while the ML-based methods instead exploit the GPU. From the table, it is easy to note that some traditional methods, such as, BT-H, BDSD-PC, MTF-GLP-HPM-R, and MTF-GLP-FS, run very fast, even though these methods are tested on the CPU. Other traditional approaches, *i.e.,* SR-D and TV, instead take more time (in particular, the TV). The testing times of the ML-based methods are quite close (less than 1 second) to the very fast traditional techniques. This is because ML approaches take advantages of the use of the GPU.

3) Training Time, Parameter Amount, and GFLOPs: We also investigate the training times of all the ML-based methods to evaluate the cost of the training. From the first row in Tab. XIX, it is clear that the slowest method, *i.e.,* BDPN, needs almost one day to train the network on the WV3 training dataset, while the fastest approach, *i.e.,* PanNet, can complete the training phase in two hours. Looking at the parameter amount (second row in Tab. XIX), BDPN has the highest value, instead, DiCNN gets the lowest one. Finally, by evaluating the giga floating point operations per second (GFlops), BDPN and DiCNN show the extreme values, again.

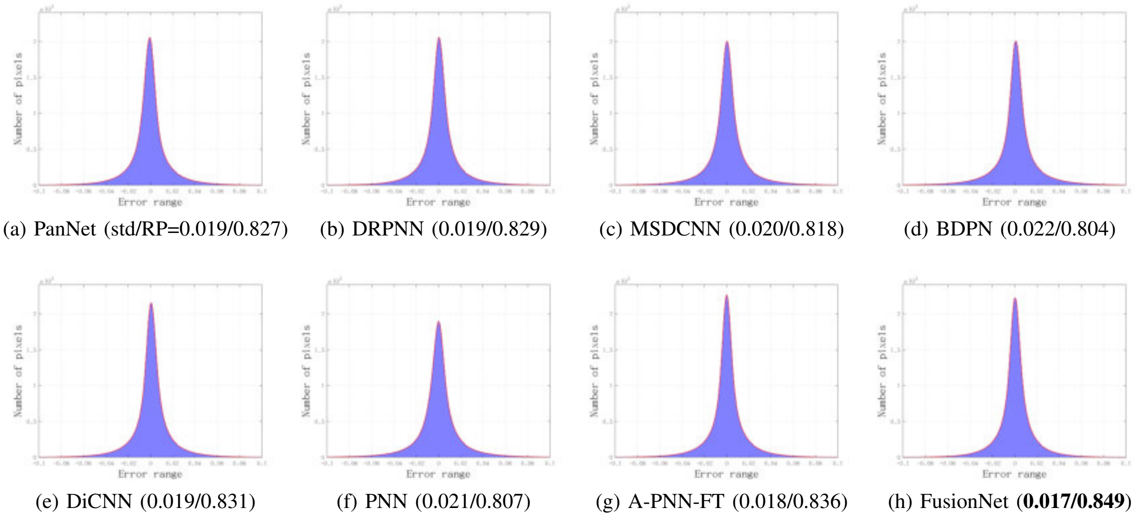
4) Histogram Comparison of Error Maps: Fig. 19 shows the histograms of the errors between each fused image and the GT evaluated on 4 reduced resolution WV3 datasets, also used in Sect. IV-G1. From the figure, we can see that the standard deviations of the A-PNN-FT and the FusionNet get the smaller results showing better overall results for this test case. Moreover, the range proportion (RP) within $[-0.02,\ 0.02]$ (the larger RP, the better performance) has also been reported in Fig. 19. Again, the best values are obtained by the FusionNet and the A-PNN-FT.



(a) PanNet (b) DRPNN (c) MSDCNN (d) BDPN

(e) DiCNN (f) PNN (g) A-PNN-FT (h) FusionNet

Fig. 18: The convergence curves for all the compared ML-based methods. The corresponding loss functions are reported in Tab. III.



(a) PanNet (std/RP=0.019/0.827) (b) DRPNN (0.019/0.829) (c) MSDCNN (0.020/0.818) (d) BDPN (0.022/0.804)

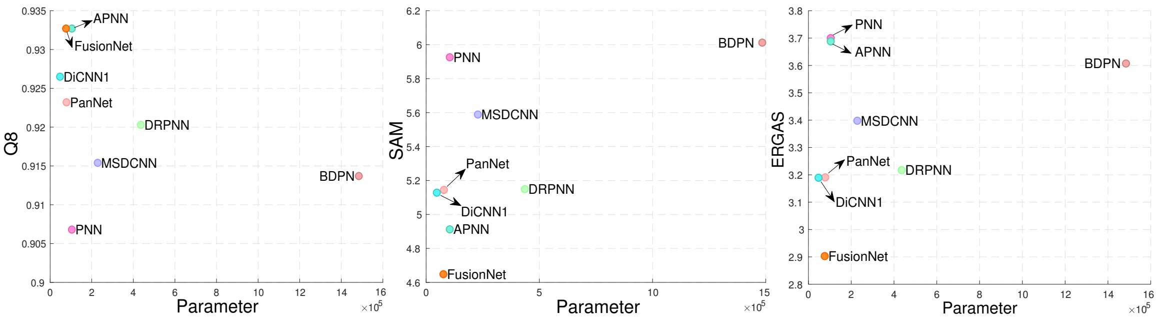
(e) DiCNN (0.019/0.831) (f) PNN (0.021/0.807) (g) A-PNN-FT (0.018/0.836) (h) FusionNet (**0.017/0.849**)

Fig. 19: The comparison of the error histogram for all the ML-based methods. The error is computed between each fused image and the GT on 4 reduced resolution WV3 datasets, also used in Sect. IV-G1. Synthetic indexes as the standard deviation (*std*) and the range proportion (RP) are reported. Best results are in boldface.

5) Performance Vs. Parameter Amount: Fig. 20 investigates the relationship between quantitative performance and parameter amount, aiming to illustrate the effectiveness of the compared ML-based methods. Again, 4 reduced resolution datasets acquired by the WV3 sensor, also used in Sect. IV-G1, have been exploited. The quality is measured using the three quality indexes at reduced resolution (*i.e.,* the Q8, the ERGAS, and the SAM). Optimal results are plotted in the top-left area for the Q8, which means getting high values of the Q8 with few parameters. Instead, for the ERGAS and the SAM, the optimal area is located in the bottom-left part of the plot. The more the methods are close to the optimal areas, the better the trade-off between quality and computational burden. Having a look at Fig. 20, we can note that the A-PNN-FT and the FusionNet get excellent performance on the data used in this analysis for all the three quality metrics.

V. Concluding Remarks

In this paper, we presented the first critical comparison among pansharpening approaches based on the ML paradigm. A complete review of the ML literature has been proposed first. Then, eight state-of-the-art solutions for sharpening MS images using PAN data have been compared. To this aim, a toolbox exploiting a common software platform and open-source ML library for all the ML approaches has been developed. All the ML approaches have been reimplemented



(a) (b) (c)

Fig. 20: The comparison of quantitative performance Vs. parameter amount on 4 reduced resolution WV3 datasets, also used in Sect. IV-G1.

TABLE XVII: Quantitative comparison of the outcomes of the benchmark on the full resolution IKONOS Toulouse dataset, see also Fig. 11. The ML-based approaches are trained on the QB dataset. Bold: the best among all the compared methods; Underline: the best among all the ML-based methods.

	D_λ	D_S	**HQNR**
CS/MRA/VO			
EXP	0.0560	0.1723	0.7813
BT-H	0.0690	0.0812	0.8554
BDSD-PC	0.0880	0.0617	0.8557
C-GSA	0.0641	0.0371	0.9012
SR-D	**0.0163**	0.0522	0.9323
MTF-GLP-HPM-R	0.0275	0.0853	0.8896
MTF-GLP-FS	0.0285	0.0908	0.8834
TV	0.0472	0.0307	0.9235
ML			
PanNet	0.0239	0.0344	0.9425
DRPNN	0.0723	0.0197	0.9095
MSDCNN	0.0830	0.0261	0.8930
BDPN	0.0699	0.0405	0.8924
DiCNN	0.1507	**0.0156**	0.8361
PNN	0.0823	0.0453	0.8761
A-PNN-FT	0.0282	0.0194	**0.9529**
FusionNet	0.0662	0.0263	0.9093

exploiting the common software platform (we selected Pytorch to this aim). The developed toolbox will be freely distributed to the community. A careful tuning phase has been performed to ensure the highest performance for each one of the compared approaches. A broad experimental analysis, exploiting different test cases, has been conducted with the aim of assessing the performance of each ML-based state-of-the-art approach. Widely used sensors for pansharpening have been involved (*i.e.,* WorldView-2, WorldView-3, WorldView-4, QuickBird, and IKONOS). The assessments both at reduced resolution and at full resolution have been considered. The comparison among ML-based approaches has also been enlarged to state-of-the-art methods belonging to different paradigms (*i.e.,* CS, MRA, and VO). The generalization ability of the networks with respect to the changes of the acquisition sensor and scenario has also been reported. Finally, a wide computational analysis has been presented in the discussions section of the paper.

ML-based approaches have demonstrated their outstanding performance in scenarios close to the ones presented during the training phase. Instead, reduced performance (in particular, in comparison with recent state-of-the-art traditional methods) has been remarked when a completely different scenario is used in the testing phase, thus showing a limited generalization ability of these approaches. However, the fine-tuning strategy has proven its ability in contrasting the above-mentioned issue, guaranteeing high performance even in these challenging test cases. The computational burden, measured during the testing phase, of the compared ML approaches can be considered adequate, even in comparison with the fastest traditional methods. Anyway, the training phase is still time consuming for several approaches requiring even one day (see the BDPN case) for the training with a relative small amount of samples.

Finally, we want to draw some guidelines for the developments of new ML-based pansharpening approaches. Indeed, focusing on the analyzed ML-based pansharpening approaches, it can be remarked that the skip connection operation can help ML-based methods in getting a faster convergence. Instead, the design of multiscaled architectures (even including the bidirectional structure) can support a better extraction and learning of the features. Furthermore, the fine-tuning technique and the learning in a specific domain (*i.e.*, not in the original image domain) can increase the generalization ability of the networks.

However, some challenges still exist representing room for improvement for researches in the next future. Specifically, as already pointed out above, the computational burden is still an open issue pushing researchers in developing networks with a reduced parameters amount (even getting a fast convergence) while taking care of the network's effectiveness. Moreover, the generalization ability is limited for most of the new developments in ML for pansharpening. This is a crucial point to be addressed to move towards the use of machine learning products for remote sensing image fusion in a commercial environment. Finally, the original idea of working at reduced

TABLE XVIII: The comparison of testing times (seconds) for all the compared methods. Note that the traditional methods (first row) are implemented on CPU and the ML-based approaches (second row) instead exploit the GPU. The times are computed on 4 reduced resolution WV3 testing datasets.

	EXP	**BT-H**	**BDSD-PC**	**C-GSA**	**SR-D**	**MTF-GLP-HPM-R**	**MTF-GLP-FS**	**TV**
Testing Time	**0.007**	0.092	0.234	1.305	7.138	0.246	0.314	31.232
	PanNet	**DRPNN**	**MSDCNN**	**BDPN**	**DiCNN**	**PNN**	**A-PNN-FT**	**FusionNet**
Testing Time	0.339	0.337	0.442	0.493	0.370	0.456	0.921	0.376

TABLE XIX: The comparison of training times (Hours: Minutes), parameter amount, and GFlops for all the compared ML-based methods. The WV3 training dataset is used as reference for this evaluation.

	PanNet	**DRPNN**	**MSDCNN**	**BDPN**	**DiCNN**	**PNN**	**A-PNN-FT**	**FusionNet**
Train. Time	**1:46**	4:42	3:08	23:22	8:21	8:40	7:55	3:01
Para. #	78,504	433,465	228,556	1,484,412	**47,369**	104,360	104,360	76,308
GFlops	0.32	1.78	0.91	3.80	**0.19**	0.29	0.22	0.32

resolution to get labels to train networks is helpful. However, it is based on the hypothesis of "invariance among scales" that could not be valid. Thus, as already pointed out in our literature review in Sect. I, new (unsupervised) approaches based on loss functions measuring similarities at full resolution have been developed. This is an interesting research line but future developments are still required, even considering the need of new studies about more accurate quality metrics at full resolution.

References

[1] L. Alparone, B. Aiazzi, S. Baronti, and A. Garzelli, *Remote Sensing Image Fusion*, CRC Press, Boca Raton, FL, Jan. 2015.

[2] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. Licciardi, R. Restaino, and L. Wald, "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2565–2586, May. 2015.

[3] P. Ghamisi, B. Rasti, N. Yokoya, Q. Wang, B. Hofle, L. Bruzzone, F. Bovolo, M. Chi, K. Anders, R. Gloaguen, P. M. Atkinson, and J. A. Benediktsson, "Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, pp. 6–39, Mar. 2019.

[4] G. Vivone, M. Dalla Mura, A. Garzelli, R. Restaino, G. Scarpa, M. O. Ulfarsson, L. Alparone, and J. Chanussot, "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.

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

[6] S. Li, X. Kang, L. Fang, J. Hu, and H. Yin, "Pixel-level image fusion: A survey of the state of the art," *Inform. Fusion*, vol. 33, pp. 100–112, Jan. 2017.

[7] X. Meng, H. Shen, H. Li, L. Zhang, and R. Fu, "Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges," *Inform. Fusion*, vol. 46, pp. 102–113, Mar. 2019.

[8] W. Carper, T. Lillesand, and R. Kiefer, "The use of intensity-hue-saturation transformations for merging SPOT panchromatic and multispectral image data," *Photogramm. Eng. Remote Sens.*, vol. 56, pp. 459–467, Apr. 1990.

[9] P. S. Chavez Jr., S. C. Sides, and J. A. Anderson, "Comparison of three different methods to merge multiresolution and multispectral data: Landsat TM and SPOT panchromatic," *Photogramm. Eng. Remote Sens.*, vol. 57, pp. 295–303, Mar. 1991.

[10] P. S. Chavez Jr. and A. W. Kwarteng, "Extracting spectral contrast in Landsat Thematic Mapper image data using selective principal component analysis," *Photogramm. Eng. Remote Sens.*, vol. 55, pp. 339–348, Jan. 1989.

[11] V. K. Shettigara, "A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set," *Photogramm. Eng. Remote Sens.*, vol. 58, pp. 561–567, 1992.

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

[13] B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pansharpening through multivariate regression of MS+Pan data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, pp. 3230–3239, Oct. 2007.

[14] G. A. Licciardi, M. M. Khan, J. Chanussot, A. Montanvert, L. Condat, and C. Jutten, "Fusion of hyperspectral and panchromatic images using multiresolution analysis and nonlinear PCA band reduction," *EURASIP J. Adv. Signal Process.*, vol. 2012, pp. 207, Sept. 2012.

[15] J. Choi, K. Yu, and Y. Kim, "A new adaptive component-substitution-based satellite image fusion by using partial replacement," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, pp. 295–309, Jan. 2011.

[16] R. Restaino, M. Dalla Mura, G. Vivone, and J. Chanussot, "Context-adaptive pansharpening based on image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 753–766, Feb. 2017.

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

[18] G. Vivone, "Robust band-dependent spatial-detail approaches for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, pp. 6421–6433, Sept. 2019.

[19] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Commun. Lett.*, vol. COM–31, no. 4, pp. 532–540, Apr. 1983.

[20] G. P. Nason and B. W. Silverman, "The stationary wavelet transform and some statistical applications," in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, Eds., vol. 103, pp. 281–299. Springer-Verlag, New York, NY, USA, 1995.

[21] J. L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.

[22] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[23] G. Vivone, R. Restaino, G. Licciardi, M. Dalla Mura, and J. Chanussot, "Multiresolution analysis and component substitution techniques for hyperspectral pansharpening," in *Proc. IEEE IGARSS*, Jul. 2014, pp. 2649–2652.

[24] L. Alparone, S. Baronti, B. Aiazzi, and A. Garzelli, "Spatial methods for multispectral pansharpening: Multiresolution analysis demystified," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2563–2576, May. 2016.

[25] S. Zheng, W. Shi, J. Liu, and J. Tian, "Remote sensing image fusion using multiscale mapped LS-SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1313–1322, May. 2008.

[26] R. Restaino, G. Vivone, M. Dalla Mura, and J. Chanussot, "Fusion of multispectral and panchromatic images based on morphological

operators," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2882–2895, Jun. 2016.

[27] G. Vivone, L. Alparone, A. Garzelli, and S. Lolli, "Fast reproducible pansharpening based on instrument and acquisition modeling: AWLP revisited," *Remote Sensing*, vol. 11, no. 19, pp. 2315:1–2315:23, Oct. 2019.

[28] R. Restaino, G. Vivone, P. Addesso, and J. Chanussot, "A pansharpening approach based on multiple linear regression estimation of injection coefficients," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 102–106, Jan. 2020.

[29] Y. Zhang, "A new merging method and its spectral and spatial effects," *Int. J. Remote Sens.*, vol. 20, no. 10, pp. 2003–2014, 1999.

[30] S. Lolli, L. Alparone, A. Garzelli, and G. Vivone, "Haze correction for contrast-based multispectral pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2255–2259, Dec. 2017.

[31] 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.

[32] G. Vivone, M. Simões, M. Dalla Mura, R. Restaino, J.M. Bioucas-Dias, G.A. Licciardi, and J. Chanussot, "Pansharpening based on semiblind deconvolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1997–2010, Apr. 2015.

[33] G. Vivone, P. Addesso, R. Restaino, M. Dalla Mura, and J. Chanussot, "Pansharpening based on deconvolution for multiband filter estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 540–553, Jan. 2019.

[34] G. Vivone and J. Chanussot, "Fusion of short-wave infrared and visible near-infrared WorldView-3 data," *Inform. Fusion*, vol. 61, pp. 71–83, Sept. 2020.

[35] L. Alparone, A. Garzelli, and G. Vivone, "Inter-sensor statistical matching for pansharpening: Theoretical issues and practical solutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4682–4695, Aug. 2017.

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

[37] M. Ghahremani and H. Ghassemian, "Remote-sensing image fusion based on curvelets and ICA," *Int. J. Remote Sens.*, vol. 36, no. 16, pp. 4131–4143, Nov. 2015.

[38] V. P. Shah, N. H. Younan, and R. L. King, "An efficient pan-sharpening method via a combined adaptive-PCA approach and contourlets," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1323–1335, May. 2008.

[39] W. Liao, X. Huang, F. Van Coillie, G. Thoonen, A. Pižurica, P. Scheunders, and W. Philips, "Two-stage fusion of thermal hyperspectral and visible RGB image by PCA and guided filter," in *Proc. IEEE WHISPERS*, Jun. 2015, pp. 1–4.

[40] P. Liu, L. Xiao, and T. Li, "A variational pan-sharpening method based on spatial fractional-order geometry and spectral-spatial low-rank priors," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, Mar. 2018.

[41] L. J. Deng, G. Vivone, W. Guo, M. Dalla Mura, and J. Chanussot, "A variational pansharpening approach based on reproducible kernel hilbert space and heaviside function," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4330–4344, Sept. 2018.

[42] C. H. Wang, C. H. Lin, J. Bioucas-Dias, W. C. Zheng, and K. H. Tseng, "Panchromatic sharpening of multispectral satellite imagery via an explicitly defined convex self-similarity regularization," in *Proc. IEEE IGARSS*, Nov. 2019, pp. 3129–3132.

[43] T. Wang, F. Fang, F. Li, and G. Zhang, "High-quality Bayesian pansharpening," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 227–239, Aug. 2019.

[44] L. J. Deng, M. Feng, and X. C. Tai, "The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-Laplacian prior," *Inform. Fusion*, vol. 52, pp. 76–89, Dec. 2019.

[45] R. Dian, S. Li, B. Sun, and A. Guo, "Recent advances and new guidelines on hyperspectral and multispectral image fusion," *Inform. Fusion*, vol. 69, pp. 40–51, May. 2021.

[46] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5135–5146, Oct. 2019.

[47] Z. C. Wu, T. Z. Huang, L. J. Deng, G. Vivone, J. Q. Miao, J. F. Hu, and X. L. Zhao, "A new variational approach based on proximal deep injection and gradient intensity similarity for spatio-spectral image fusion," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 13, pp. 6277–6290, Oct. 2020.

[48] J. Duran, A. Buades, B. Coll, C. Sbert, and G. Blanchet, "A survey of pansharpening methods with a new band-decoupled variational model," *ISPRS J. Photogramm. Remote Sensing*, vol. 125, pp. 78–105, Mar. 2017.

[49] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé, "A variational model for P+XS image fusion," *Int. J. Comput. Vision*, vol. 69, no. 1, pp. 43–58, Aug. 2006.

[50] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "A new pansharpening algorithm based on total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 318–322, Jan. 2014.

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

[52] H. A. Aly and G. Sharma, "A regularized model-based optimization framework for pan-sharpening," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2596–2608, Apr. 2014.

[53] M. Möller, T. Wittman, A. L. Bertozzi, and M. Burger, "A variational approach for sharpening high dimensional images," *SIAM J. Imaging Sci.*, vol. 5, pp. 150–178, Jan. 2012.

[54] G. Zhang, F. Fang, A. Zhou, and F. Li, "Pan-sharpening of multi-spectral images using a new variational model," *Int. J. Remote Sens.*, vol. 36, pp. 1484–1508, Mar. 2015.

[55] F. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Model-based reduced-rank pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 656–660, Apr. 2020.

[56] D. Fasbender, J. Radoux, and P. Bogaert, "Bayesian data fusion for adaptable image pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1847–1857, Jun. 2008.

[57] Y. Zhang, S. De Backer, and P. Scheunders, "Noise-resistant wavelet-based bayesian fusion of multispectral and hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3834–3843, Nov. 2009.

[58] F. Palsson, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Model-based fusion of multi- and hyperspectral images using PCA and wavelets," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2652–2663, May. 2015.

[59] Y. Zhang, A. Duijster, and P. Scheunders, "A Bayesian restoration approach for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, pp. 3453–3462, Sept. 2012.

[60] S. Li and B. Yang, "A new pan-sharpening method using a compressed sensing technique," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, pp. 738–746, Feb. 2011.

[61] C. Jiang, H. Zhang, H. Shen, and L. Zhang, "A practical compressed sensing-based pan-sharpening method," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, pp. 629–633, Jul. 2015.

[62] S. Li, H. Yin, and L. Fang, "Remote sensing image fusion via sparse representations over learned dictionaries," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, pp. 4779–4789, Sept. 2013.

[63] M. Cheng, C. Wang, and J. Li, "Sparse representation based pansharpening using trained dictionary," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, pp. 293–297, Jan. 2014.

[64] X. X. Zhu and R. Bamler, "A sparse image fusion algorithm with application to pan-sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, pp. 2827–2836, May. 2013.

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

[66] M. R. Vicinanza, R. Restaino, G. Vivone, M. Dalla Mura, G. Licciardi, and J. Chanussot, "A pansharpening method based on the sparse representation of injected details," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 180–184, Jan. 2015.

[67] X. Tian, Y. Chen, C. Yang, X. Gao, and J. Ma, "A variational pansharpening method based on gradient sparse representation," *IEEE Singal Proc. Let.*, vol. 27, pp. 1180–1184, Jul. 2020.

[68] W. Huang, L. Xiao, Z. Wei, H. Liu, and S. Tang, "A new pan-sharpening method with deep neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1037–1041, May. 2015.

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

[70] J. Zhong, B. Yang, G. Huang, F. Zhong, and Z. Chen, "Remote sensing image fusion with convolutional neural network," *Sensing and Imaging*, vol. 17, no. 10, pp. 1–16, Jun. 2016.

[71] 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, pp. 1795–1799, Oct. 2017.

[72] Y. Rao, L. He, and J. Zhu, "A residual convolutional neural network for pan-sharpening," in *Proc. RSIP*, May. 2017, pp. 1–4.

[73] J. Yang, X. Fu, Y. Hu, Y. Huang, and J. Paisley, "PanNet: A deep network architecture for pan-sharpening," in *Proc. ICCV*, Oct 2017.

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

[75] 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.*, vol. 11, pp. 978–989, Mar. 2018.

[76] Z. Shao and J. Cai, "Remote sensing image fusion with deep convolutional neural network," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 11, no. 5, pp. 1656–1669, May. 2018.

[77] W. Yao, Z. Zeng, C. Lian, and H. Tang, "Pixel-wise regression using U-Net and its application on pansharpening," *Neurocomputing*, vol. 312, pp. 364–371, Oct. 2018.

[78] X. Liu, Y. Wang, and Q. Liu, "PSGAN: A generative adversarial network for remote sensing image pan-sharpening," in *Proc. ICIP*, Sept. 2018, pp. 873–877.

[79] Y. Zhang, C. Liu, M. Sun, and Y. Ou, "Pan-sharpening using an efficient bidirectional pyramid network," *IEEE Trans. Geosci. Remote*, vol. 57, pp. 1–15, Aug. 2019.

[80] K. Li, W. Xie, Q. Du, and Y. Li, "DDLPS: Detail-based deep laplacian pansharpening for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8011–8025, Oct. 2019.

[81] H. Zhang and J. Ma, "GTP-PNet: A residual learning network based on gradient transformation prior for pansharpening," *ISPRS J. Photogramm.*, vol. 172, pp. 223–239, Feb. 2021.

[82] J. Liu, Y. Feng, C. Zhou, and C. Zhang, "PWNet: An adaptive weight network for the fusion of panchromatic and multispectral images," *Remote Sens.*, vol. 12, pp. 2804, Aug. 2020.

[83] J. Liu, C. Zhou, R. Fei, C. Zhang, and J. Zhang, "Pansharpening via neighbor embedding of spatial details," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 14, pp. 4028–4042, Mar. 2021.

[84] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, "Hyperspectral image spatial super-resolution via 3D full convolutional neural network," *Remote Sens.*, vol. 9, no. 11, pp. 1139:1–1139:22, Nov. 2017.

[85] 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. Photogramm. Remote Sensing*, vol. 146, pp. 305–319, Dec. 2018.

[86] M. Gargiulo, A. Mazza, R. Gaetano, G. Ruello, and G. Scarpa, "Fast super-resolution of 20 m Sentinel-2 bands using convolutional neural networks," *Remote Sens.*, vol. 11, no. 22, pp. 2635:1–2635:18, Nov. 2019.

[87] Y. Qu, H. Qi, and C. Kwan, "Unsupervised sparse Dirichlet-Net for hyperspectral image super-resolution," in *Proc. CVPR*, Jun. 2018, pp. 2511–2520.

[88] Q. Xie, M. Zhou, Q. Zhao, Z. Xu, and D. Meng, "MHF-Net: An interpretable deep network for multispectral and hyperspectral image fusion," *IEEE Trans. Pattern Anal.*, 2020, doi:10.1109/TPAMI.2020.3015691.

[89] J. F. Hu, T. Z. Huang, L. J. Deng, T. X. Jiang, G. Vivone, and J. Chanussot, "Hyperspectral image super-resolution via deep spatio-spectral attention convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, doi:10.1109/TNNLS.2021.3084682.

[90] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[91] L. He, Y. Rao, J. Li, J. Chanussot, A. Plaza, J. Zhu, and B. Li, "Pansharpening via detail injection based convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 12, pp. 1188–1204, Apr. 2019.

[92] L. J. Deng, F. Vivone, C. Jin, and J. Chanussot, "Detail injection-based deep convolutional neural networks for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, 2020, doi: 10.1109/TGRS.2020.3031366.

[93] R. Dian, S. Li, and X. Kang, "Regularizing hyperspectral and multispectral image fusion by CNN denoiser," *IEEE Trans. Neur. Net. Lear.*, vol. 32, no. 3, pp. 1124–1135, Mar. 2021.

[94] Huanfeng Shen, Menghui Jiang, Jie Li, Qiangqiang Yuan, Yanchong Wei, and Liangpei Zhang, "Spatial–spectral fusion by combining deep learning and variational model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 6169–6181, 2019.

[95] Weiying Xie, Jie Lei, Yuhang Cui, Yunsong Li, and Qian Du, "Hyperspectral pansharpening with deep priors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1529–1543, 2020.

[96] Z. C. Wu, T. Z. Huang, L. J. Deng, J. F. Hu, and G. Vivone, "VO+Net: An adaptive approach using variational optimization and deep learning for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, 2021, doi:10.1109/TGRS.2021.3066425.

[97] Y. Feng, J. Liu, K. Chen, B. Wang, and Z. Zhao, "Optimization algorithm unfolding deep networks of detail injection model for pansharpening," *IEEE Geosci. Remote Sens. Lett.*, 2021, doi:10.1109/LGRS.2021.3077183.

[98] Shuang Xu, Jiangshe Zhang, Zixiang Zhao, Kai Sun, Junmin Liu, and Chunxia Zhang, "Deep gradient projection networks for pan-sharpening," in *Proc. CVPR*, 2021, pp. 1366–1375.

[99] Xiangyong Cao, Xueyang Fu, Danfeng Hong, Zongben Xu, and Deyu Meng, "Pancsc-net: A model-driven deep unfolding method for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2021.

[100] Haitao Yin, "Pscsc-net: A deep coupled convolutional sparse coding network for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2021.

[101] 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," *Inform. Fusion*, vol. 62, pp. 110–120, Oct. 2020.

[102] Shuyue Luo, Shangbo Zhou, Yong Feng, and Jiangan Xie, "Pansharpening via unsupervised convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4295–4310, 2020.

[103] Y Qu, R Baghbaderani, H Qi, and C Kwan, "Unsupervised pansharpening based on self-attention mechanism," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3192–3208, 2021.

[104] Matteo Ciotola, Sergio Vitale, Antonio Mazza, Giovanni Poggi, and Giuseppe Scarpa, "Pansharpening by convolutional neural networks in the full resolution framework," 2021.

[105] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," 2014.

[106] Zhimin Shao, Zexin Lu, Maosong Ran, Leyuan Fang, Jiliu Zhou, and Yi Zhang, "Residual encoder–decoder conditional generative adversarial network for pansharpening," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1573–1577, 2020.

[107] Wenqian Dong, Shaoxiong Hou, Song Xiao, Jiahui Qu, Qian Du, and Yunsong Li, "Generative dual-adversarial network with spectral fidelity and spatial enhancement for hyperspectral pansharpening," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[108] Zixiang Zhao, Jiangshe Zhan, Shuang Xu, Kai Sun, Lu Huang, Junmin Liu, and Chunxia Zhang, "Fgf-gan: A lightweight generative adversarial network for pansharpening via fast guided filter," in *Proc. ICME*, 2021, pp. 1–6.

[109] Weiying Xie, Yuhang Cui, Yunsong Li, Jie Lei, Qian Du, and Jiaojiao Li, "Hpgan: Hyperspectral pansharpening using 3-d generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 463–477, 2021.

[110] Anais Gastineau, Jean-Francois Aujol, Yannick Berthoumieu, and Christian Germain, "Generative adversarial network for pansharpening with spectral and spatial discriminators," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

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

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

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

[114] G. Vivone, R. Restaino, M. Dalla Mura, G. Licciardi, and J. Chanussot, "Contrast and error-based fusion schemes for multispectral image

pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 5, pp. 930–934, May. 2014.

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

[116] B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pansharpening through multivariate regression of MS+Pan data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3230–3239, Oct. 2007.

[117] G. Vivone, "Robust band-dependent spatial-detail approaches for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6421–6433, Sept. 2019.

[118] G. Vivone, R. Restaino, and J. Chanussot, "A regression-based high-pass modulation pansharpening approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 984–996, Feb. 2018.

[119] G. Vivone, R. Restaino, and J. Chanussot, "Full scale regression-based injection coefficients for panchromatic sharpening," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3418–3431, Jul. 2018.

[120] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[121] 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.

[122] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. ECCV*, 2016, pp. 353–369.

[123] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, Jun. 2016, pp. 1646–1654.

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

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

[126] L. Wald, *Data Fusion: Definitions and Architectures — Fusion of images of different spatial resolutions*, Les Presses de l'École des Mines, Paris, France, 2002.

[127] A. Garzelli and F. Nencini, "Hypercomplex quality assessment of multi/hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 6, no. 4, pp. 662–665, Oct. 2009.

[128] 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.

[129] M. M. Khan, L. Alparone, and J. Chanussot, "Pansharpening quality assessment using the modulation transfer functions of instruments," *IEEE Trans. Geosci. Remote Sens.*, vol. 11, no. 47, pp. 3880–3891, Nov. 2009.