Target-adaptive CNN-based pansharpening

Giuseppe Scarpa, Sergio Vitale, and Davide Cozzolino

Abstract—We recently proposed a convolutional neural network for remote sensing image pansharpening obtaining a significant performance gain over the state of the art. In this paper, we explore a number of architectural and training variations to this baseline, achieving further performance gains with a lightweight network which trains very fast. Leveraging on this latter property, we propose a target-adaptive usage modality which ensures a very good performance also in the presence of a mismatch w.r.t. the training set, and even across different sensors. The proposed method, published online as an off-the-shelf software tool, allows users to perform fast and high-quality CNN-based pansharpening of their own target images on general-purpose hardware.

I. INTRODUCTION

Thanks to continuous technological progresses, there has been a steady improvement in the quality of remote sensing products, and especially in the spatial and spectral resolution of images. Then, when technology reaches its limits, signal processing methods may provide a further quality boost. Pansharpening is among the most successful examples of such a phenomenon. Given a high spatial resolution panchromatic band (PAN) and a low resolution multispectral stack (MS), it generates a datacube at the highest resolution in both the spectral and spatial domains. Results are already promising, but intense research is going on to approach more and more closely the quality of ideal high-resolution data.

In the last decades, many different approaches have been proposed to address the pansharpening problem. A classic approach is the component substitution (CS) [1], where the multispectral component is upsampled and transformed in a suitable domain and the panchromatic band is used to replace one of the transformed bands before inverse transform to the original domain. Under the restriction that only three bands are concerned, the Intensity-Hue-Saturation (IHS) transform can be used, with the intensity component replaced by the panchromatic band [2]. This same approach has been then generalized in [3] (GIHS) to include additional bands. Many other transforms have been considered for CS like, for example, the principal component analysis [4], the Brovey transform [5], and Gram-Schmidt (GS) decomposition [6]. More recently, Adaptive CS methods have also been introduced, like the advanced versions of GIHS and GS adopted in [7], the partial substitution method (PRACS) proposed in [8], and optimization-based techniques [9].

CS methods approach the pansharpening problem from the spectral perspective, as the fusion occurs with respect to a spectral transform. Another class of methods regard the problem from the geometric, or spatial, perspective mostly relying on multiresolution analysis (MRA) [10]. MRA-based methods aim to extract spatial details from the PAN component to be later injected in the resampled MS component. Spatial details can be extracted in different ways, using, for example, decimated or undecimated wavelet transforms [11], [10], [12], [13]. Laplacian pyramids [14], [15], [16], [17], [18], or other nonseparable transforms, like the contourlet [19].

The separation between CS and MRA methods is neither always clear-cut nor exhaustive. There are in fact many examples of methods which are better cast as statistical or variational that get state of the art results [20], [21], [22], [23], [24]. However this CS-MRA dichotomy is useful to understand the behaviour of any method falling in these categories as highlighted in [25], [26]. Specifically, given well registered MS-PAN components, and accurate modeling of the sensor Modulation Transfer Function (MTF), methods based on MRA achieve usually a better pansharpening quality than those based on CS [27]. On the contrary, when MS-PAN misregistration occurs, both CS and MRA methods may lose geometric sharpness, but the latter suffer also from spectral mismatch, making them unsuitable for applications where spectral accuracy is of critical importance [26].

In the last years machine learning methods have gained much attention from both signal processing and remote sensing communities. Compressive sensing and dictionary based methods, for example, have been successfully applied to pansharpening in several papers [28], [29], [30], [31], [32]. Very recently, following the recent technological and theoretical advances in computer vision and related fields, also deep learning methods have been applied to remote sensing problems, and several papers have been proposed to address pansharpening [33], [34], [35], [36], [37]. In particular, our CNN-based solution [34], inspired by the work on super-resolution [38], provided a significant performance improvement with respect to the previous state-of-the-art.

In this work we start from the baseline solution of [34] and explore a number of variations aimed at improving both performance and robustness, including different learning strategies, cost functions, and architectural choices. The most remarkable improvements are obtained by including a target-adaptive tuning phase, which solves to a large extent the problem of insufficient training data, allowing users to apply the proposed architecture to their own data and achieve good results consistently. The proposed solutions are extensively tested on images acquired by a number of sensors, covering different spatial and spectral resolutions. A substantial improvement is observed over both the baseline and the state-of-the-art methods available in the Open Remote Sensing repository [39], under a wide range of quality measures.

In the rest of the paper we describe the baseline method (Section II), the proposed architectural improvements (Sec-
Section III), the target adaptive solution (Section IV), and the experimental results (Section V), before drawing conclusions (Section VI).

II. A pansharpening Neural Network

In [34], [40], inspired to previous work on super-resolution [38], we proposed the Pansharpening Neural Network (PNN), summarized in the block diagram of Fig. 1. The core of the network is a simple three-layer convolutional neural network (CNN), not shown here for brevity. The CNN takes in input the panchromatic band \( x_{\text{PAN}} \) (blue), the multispectral (MS) component \( x_{\text{MS}} \) (red) up-sampled via polynomial interpolation, and a few radiometric indices \( x_F \) (green) extracted from the MS component and interpolated as well. The last component, comprising some nonlinear combination of MS spectral bands, has proven experimentally to improve the network performance.

The CNN is composed by three convolutional layers with rectified linear unit (ReLU) activations in both the input and the hidden layers, and linear activation in the output layer. For each layer, assuming \( N \) input bands, \( M \) output bands, and filters with \( K \times K \) receptive field (spatial support), a number of parameters must be learned on the training set, a \( M \times [N \times (K \times K)] \) tensor, \( w \), accounting for the weights, and a \( M \)-vector, \( b \), for the biases. For layer \( l \), with input \( x^{(l)} \), the filter output is computed as

\[
z^{(l)} = w^{(l)} \ast x^{(l)} + b^{(l)},
\]

where the \( m \)-th component can be expressed as

\[
z^{(l)}(m, \cdot, \cdot) = \sum_{n=1}^{N} w^{(l)}(m, n, \cdot, \cdot) \ast x^{(l)}(n, \cdot, \cdot) + b^{(l)}(m)
\]

in terms of the usual 2D convolution. After filtering, in the input and the hidden layers a pointwise nonlinear function is also applied, in particular a Rectified Linear Unit, ReLU(\( \cdot \)) = \( \max(0, \cdot) \), to obtain the actual feature maps \( y^{(l)} \)

\[
y^{(l)} = f_l(x^{(l)}, \Phi^l) = \max(0, z^{(l)})
\]

where \( \Phi^l \triangleq (w^{(l)}, b^{(l)}) \). The choice of ReLU nonlinearities is motivated experimentally [41] by the good convergence properties they guarantee. Notice that, as neither stride nor pooling are used, each layer preserves the input resolution and hence \( x^{(l)} \) and \( y^{(l)} \) have the same spatial size.

A. Learning

Let \( x = (x_{\text{PAN}}, x_{\text{MS}}, x_F) \) denote the composite CNN input stack and \( f(x, \Phi) \) be the overall function computed by the CNN, with \( \Phi = (\Phi_1, \Phi_2, \Phi_3) \) the collection of its parameters. In order to learn the network parameters, reference data are required, that is, examples of perfectly pansharpened images coming from the same sensor for which the network is designed. Unfortunately, images at full spatial and spectral resolution are not available, which complicates training and performance assessment alike. For the purpose of validation, this problem is often addressed by resorting to the Wald protocol [42], which consists in downsampling both the PAN and the MS components, so that the original MS component can be taken as a reference for the pansharpening of the downsampled data. In [42] the selection of the low-pass filter to apply before downsampling, highly sensor-dependent, is left as an open problem. This is a key issue, impacting on the consistency of quality assessment across scales, and hence on the correctness of the whole validation protocol. In [16] it is proposed to use suitable approximations of the sensor modulation transfer functions (MTF) to this end. This solution was adopted in [34] not only for quality assessment but also for the purpose of learning.

Fig. 2 shows how the Wald’s protocol is used to create training examples. This scheme generalizes readily to the case where additional low-resolution input bands are considered, like the MS radiometric indices. As different sensors have different MTFs and, more in general, the PAN and MS components correlate in different ways, the resulting network is strictly sensor-dependent. It is also clear that a wrong modeling of the sensor MTF may impact severely on the network performance.

Given a training set \( \chi = \{x_1, \ldots, x_Q\} \), generated by the Wald’s protocol with any MTF modeling, comprising \( Q \) input-output image pairs \( x \triangleq (x, x_{\text{ref}}) \), the objective of the training phase is to find

\[
\Phi = \arg\min_{\Phi} J(\chi, \Phi) \triangleq \arg\min_{\Phi} \frac{1}{Q} \sum_{x \in \chi} L(x, \Phi)
\]

where \( L(x, \Phi) \) is a suitable loss function.

In [34] the mean square error (MSE) was used as loss function:

\[
L(x, \Phi) \propto \|f(x, \Phi) - x_{\text{ref}}\|_2^2
\]

and the minimization was carried out by stochastic gradient descent (SGD) with momentum [43]. In particular, the training set is partitioned in \( P \) batches, \( \{B_1, \ldots, B_P\} \), with \( \bigcup_j B_j = \chi \), and at each iteration a batch is sampled and used to estimate the gradient and update parameters as

\[
\nu^{(n+1)} \leftarrow \mu \nu^{(n)} + \alpha \nabla_{\Phi} J(B_{j_n}, \Phi^{(n)})
\]

\[
\Phi^{(n+1)} \leftarrow \Phi^{(n)} - \nu^{(n+1)}
\]

A whole scan of the training set is called an epoch, and training a deep network may require thousands of epochs. Training
efficiency and accuracy depend heavily on the algorithm hyperparameters, learning rate, \( \alpha \), momentum, \( \mu \), and velocity, \( \nu \), the most critical being the learning rate, which can cause instability when too large or slow convergence when too small. In [34], after extensive experiments, the learning rate was set to \( 10^{-4} \) for \( \Phi_1 \) and \( \Phi_2 \) and to \( 10^{-5} \) for \( \Phi_3 \).

III. IMPROVING CNN-BASED PANSHARPENING

Although the PNN architecture relies on solid conceptual foundations, following a well-motivated path from dictionary-based super-resolution [44], to its CNN-based counterpart [38], and finally to pansharpening, there is plenty of room for variations and, possibly, further improvements. Therefore we explored experimentally a number of alternative architectures and learning modalities. In the following, we describe only the choices that led to significant improvements or are otherwise worth analyzing, that is,
- using L1 loss;
- working on image residuals;
- using deeper architectures.

A. Using L1 loss function

In deep learning, choosing the “right” loss function can make the difference between being stuck with disappointing results and achieving the desired output. With pansharpening there is no shortage of candidate loss functions [25, 44] but most of them are quite complicated, and end up being time-consuming and sometimes unstable. Especially the first reason motivates us to consider only Ln norms. In [34] we used the L2 loss, as in [38], but our preliminary experiments proved the L1 norm to be a much better choice. Surprisingly, by training the network to minimize a L1 loss, we achieved better results even in terms of mean square error (MSE) or other L2 related indicators. This behaviour is in general possible because of the non convexity of the target and, indeed, has already been observed and discussed in the deep learning literature [45]. On one hand, when the regression targets are unbounded, training with L2 loss requires careful tuning of learning rates in order to prevent exploding gradients. On the other hand, and probably more important, the L2 norm penalizes heavily large errors but is less sensitive to small errors, which means that the learning process slows down significantly as the output approaches the objective. This is the working point of highest interest for our application, because the quality of CNN-based pansharpening is already very good, according to both numerical indicators and visual inspection [34]. To achieve further improvements, one must focus on small errors, a goal for which the L1 norm is certainly more fit. On the down side, the L1 norm is more prone to instabilities but in our experiments these never prevented eventual convergence and satisfactory results.

B. Working on image residuals

In the baseline architecture proposed in [34], the network is trained to reconstruct the whole target image. However, the low-pass component of the output, that is, the up-sampled MS component, is already available and only the high-resolution residual need be generated. Based on this observation, we modified the network to let it reconstruct only the missing part of the desired output.

The residual-based version of our baseline solution is shown in Fig.3 where the preprocessing is omitted for the sake of simplicity. The core CNN is trained to generate only the residual component, namely the desired pansharpened image minus its low-pass component. Therefore, the desired output is obtained by summing the up-sampled MS component, \( x_{MS} \), made available through a skip connection, to the network output, \( f(x, \Phi) \). The loss is then computed as

\[
L(\vec{x}, \Phi) \propto ||(f(x, \Phi) + x_{MS}) - x_{ref}||_2^2
\]

with the residual reference defined as \( \Delta x_{ref} \triangleq x_{ref} - x_{MS} \) and, accordingly, the training samples redefined as \( \vec{x} \triangleq (x, \Delta x_{ref}) \). From the training perspective, the only difference with the baseline solution of Fig.1 is that reference data are obtained by subtracting the interpolated MS component \( x_{MS} \).
If we also replace the L2 norm with L1 norm as suggested in the previous subsection, the loss becomes eventually

\[
L(x, \Phi) \propto ||f(x, \Phi) - \Delta x_{\text{ref}}||_1
\]

Note that residual learning is not a new idea. In [46], it was used for dictionary-based super-resolution, proving effective both in terms of accuracy and training speed. More recently, it was advocated for training very deep CNNs [48], [49], and used successfully in various applicative fields, e.g. image denoising with very deep networks [50]. In [51] a residual-based regressor was also proposed for pansharpening. As said before, residual learning is a natural choice for pansharpening, due to the availability of the low-pass component. More in general, it was observed experimentally [49] that training the network to reproduce the desired output may be quite difficult when the output is itself very similar to the input. The process becomes much more efficient when targeting differences between input and output, that is, residuals. Therefore this applies to many image restoration and enhancement tasks, such as denoising, super-resolution, and pansharpening.

Very recently, two groups of researchers have proposed using residual learning for pansharpening [36], [35], in the context of deep or very deep CNNs, claiming some performance improvements w.r.t. PNN. However, in both cases the experimental validation is somewhat faulty, preventing solid conclusions. In [36] experiments are carried out on Landsat 7 images, with geometric and spectral characteristics very far from those of typical multiresolution images of interest. In [35], instead, the assessment involves only reduced resolution data. Therefore, in both cases there is no clue on how the methods perform on full-resolution images of interest.

C. Using deeper architectures

A generic CNN is formed by the cascade of \( L \) processing layers, hence if computes a composite function in the form

\[
f(x, \Phi) = f_L(f_{L-1}(\ldots f_1(x, \Phi_1), \ldots, \Phi_{L-1}), \Phi_L)
\]

where \( \Phi \triangleq (\Phi_1, \ldots, \Phi_L) \). Our baseline method, as well as variations considered thus far, relies on a 3-layer CNN, which can be considered a rather shallow network. On the contrary, the current trend in the literature is towards the use of deep or very deep networks. In principle, deeper networks exhibit a superior expressiveness, because more and more abstract features can be built on top of simpler ones. Moreover, it has been demonstrated [52] that the representational capability of a network grows with its dimension. On the downside, training very deep networks may require a long time and convergence is more difficult, because information does not backpropagates easily through so many layers. A number of approaches have been proposed in the recent literature to deal with this problem, based on residual learning, suitable losses and activation functions, a careful choice of hyper-parameters, and batch normalization.

Hence, we decided to test deeper CNNs for pansharpening. Following the approach of [33], we consider \( L \) identical layers, except for the input and output layers which are modified to account for the input and output shape. Filter supports are reduced to obtain composite receptive fields that have approximately the same size as in the baseline. Like in the baseline, we use ReLU activations for the input and the hidden layers, and an identity mapping in the output layer. We also include already residual learning and L1 loss in the new solution. Finally, during training, we stabilize the layers’ inputs by means of batch normalization [54], thus removing unwanted random fluctuations and speeding-up the training phase, significantly.

D. Preliminary experiments

To gain insight into the impact of the proposed improvements we carried out some preliminary experiments on our GeoEye-1 multiresolution dataset, described in detail in Section V. Performance is assessed in terms of average error on the validation dataset vs. training time. We do not use number of iterations or epochs, as their time cost varies as a function of the architecture. Both the mean square error (MSE) and the mean absolute error (MAE), are considered, left and right parts of Fig. 4, irrespective of the loss function, L2 or L1, used to train the CNN. Indeed, since there is no consensus on the ideal performance measure for pansharpening, results with two different and well-understood norms may provide some indications on robustness across other more complicated measures.

The main phenomena are quite clear, and consistent for the two cases. First of all, replacing L2 with L1 norm in the training phase provides a significant performance gain. Then, a further improvement is obtained by adopting also residual learning in combination with the L1 loss. On the contrary, residual learning has a negative impact on performance when used in combination with L2 loss (the RL curves). A possible explanation is that residual learning works on smaller inputs, not well discriminated by the L2 norm, causing a slower backpropagation of errors. Also, increasing the network depth (always with residual learning and L1 loss) does not seem to provide any benefit, as clear by the D10 curves, associated with the best architecture found by varying number of layers, filter support, and number of features per layer. Despite batch normalization, and a careful setting of learning rates, instabilities occur, and no gain is observed over the 3-layer counterpart, at convergence. Finally, input augmentation through radiometric indices, which provided an appreciable loss reduction for the baseline, proves useless for the best methods emerging from this analysis.

Interesting results emerge also in terms of training speed. Indeed, the baseline requires a long time to reach convergence, and actually the loss keeps decreasing even after 10 hours. On the contrary, the residual-L1 version achieves the same performance after just 30 minutes, and the training appears to be complete after 2 hours. Therefore, besides providing a large performance gain, the new solution cuts training times by a factor 5 or more.

Note that similar results, not shown here for brevity, have been obtained with other datasets and sensors. We also tested the coherence between full-reference quality indicators, used on downsampled data, and no-reference quality indicators,
IV. TARGET-ADAPTIVE PANSHARPENING

A basic prescription of deep learning is to train the network of interest on a large and varied dataset, representative of the data that will be processed in actual operations. This allows the network to generalize and provide a good performance also on data never seen during training. On the contrary, if the training set is too small or not varied enough, the network may overfit these data, providing a very good performance on them, and working poorly on new data. In other words, the network is desired to be robust over a wide distribution of data, although not optimal for any of them. Once the training is over, all parameters are freezed and the network is used on the targets with no further changes. This procedure is motivated by the desire to obtain a stable and predictable network and, not least, by computational issues, since training a deep network anew for each target would be a computational nightmare. However, what if such a dedicated training were feasible in real time?

We explored this opportunity, and verified that including a target-adaptive fine-tuning step in our method is computationally feasible, actually, almost transparent to the user if suitable hardware is available, and may provide huge improvements in performance whenever a mismatch occurs between training data and target image. Therefore, we propose, here, a target-adaptive version of our pansharpening network. More precisely, we use the best architecture emerging from the analyses of previous Section, a three-layer CNN (see hyperparameters in Tab. I) with residual learning and L1 loss, and train it on the available dataset. Then, at run-time, a fine-tuning step is performed on the target data, so as to provide the desired adaptation.

It is worth emphasizing the importance of such an adaptation step for pansharpening. Indeed, contrary to what happens in other fields, no large database is available to the remote sensing community for developing and testing new solutions, and one must resort to proprietary datasets, often not large and diverse as necessary. Hence, the performance of CNN-based methods on new data may happen to be much worse than expected, even worse than conventional methods, leaving the huge potential of deep learning untapped.
Also, it should be realized that this fine-tuning step is computationally light, unlike what happens with conventional training from scratch. In our experiments, adapting the proposed network to a 1280×1280 (PAN resolution) image took about 1.5 seconds on a GPU-equipped computer (GeForce GTX Titan X, Maxwell, 12GB) and 210 seconds on a general purpose CPU (Intel Xeon E5-2670 1.80GHz, 64GB). Therefore, this additional process goes totally unnoticed in the former case, and even in the latter case, the overhead is definitely affordable for most applications, and fully worth it, in view of the ensuing performance improvement. This very short processing time is readily explained, in fact

- the network is already pre-trained, and fine tuning requires a small number of iterations;
- the adaptation involves only the target data, order of magnitudes less than typical training sets;
- the selected architecture, adopting residual learning, trains already much faster than conventional CNNs, like for example the baseline PNN;

We also underline that the fine tuning phase does not require any active user involvement.

Also in this case, before turning to extensive experiments, we present some preliminary evidence of the achievable performance improvements in the various operating conditions of interest. In particular, in Fig. 6 we report the performance gain over the proposed L1-RL architecture observed on a number of different target clips when fine tuning is performed. Performance is measured both by the Q4/Q8 full-reference measure (see Section V) computed on the reduced-resolution data, and by the no-reference QNR measure mentioned before.

In the first graph, we consider a rather favourable case in which the target clips, although disjoint from the training set, are drawn from the same large image (Caserta-GeoEye-1). Obviously, the performance gain is very limited on all clips, due to the perfect alignment between training and test data. Note also that, while there is always a gain for the full-reference measure Q4, this is not the case for the QNR, underlining again the mismatch between these two classes of measures. In the second graph, we consider the more typical case in which training set (Caserta-WorldView-2) and target clips (Washington-WorldView-2) concern different images taken by the same sensor. Here, fine tuning guarantees a significant improvement of the Q8 indicator (0.1, on the average, on a unitary scale) confirming the potential benefit of this processing step. Again, the objective Q8 measure is only mildly correlated with the QNR which, in some cases, exhibits even a significant drop. The third graph illustrates a very challenging case with extreme mismatch between training set (Caserta-WorldView-2) and target clips (Adelaide-WorldView-3) since even the sensor now is different. As could be expected, the improvement achieved on Q8 through fine tuning is always substantial, almost 0.2 on the average, and significant improvements, although smaller, are observed also in terms of QNR.

These preliminary results are extremely encouraging. In case of mismatch between training and target, the fine tuning step improves significantly the objective measures of performance. The impact on the quality of full resolution images, is more controversial. This may be due to incorrect modeling of the MTF, but also to the limited ability of these measures to assess actual image quality. In general, full-resolution quality assessment is an open issue [55], [56], [57], and the most sensible way to compare different solutions is to jointly look at the reduced-resolution and full-resolution results, and never neglect visual inspection.

V. EXPERIMENTAL ANALYSIS

To assess the performance of the proposed methods we carried out a number of experiments with real-world multiresolution images, exploring a wide range of situations.

In the following subsections we
- list the methods under test, both proposed and reference;
- summarize the set of performance measures considered in the experiments, both full-reference and no-reference, discussing briefly their significance;
- describe training and test sets, and how they are combined to explore increasingly challenging cases;
- report and comment the experimental results, both numerical and visual.

A. Methods under analysis

Our baseline is the PNN proposed in [34], with the input augmented by radiometric indexes. However, since all the solutions proposed here have been implemented in Python using Theano [58], we have re-implemented in this framework also the baseline, originally developed with Caffe [59], in order
to avoid biases due to the different arithmetic precision and/or randomization. This justifies some small numerical differences w.r.t. results reported in [34]. Furthermore, we consider three variations of PNN, L1, obtained by replacing the L2 loss with L1 loss in PNN, L1-RL, adopting in addition also the residual learning architecture, and L1-RL-FT, which fine tunes the network on the target. All these methods share the same three-layer CNN, with hyper-parameters given in Tab. I. Only for the baseline, the input channels include also some radiometric indexes (red entries in Tab. I).

Besides our CNN-based methods, we consider a number of well-known conventional techniques selected because of their good performance, and specifically, PRACS [8], Indusion [13], AWLP [12], ATWT-M3 [10], MTF-GLP-HPM [15], BDIS [9], and its recent extension C-BDIS [22]. Details on these methods can be found in [25] and in [34], and obviously in the original papers. In addition we also report as a naive reference the 23-tap polynomial interpolator (denoted EXP) used by many algorithms, including ours, as initial upsample. The software used to implement the methods and carry out all experiments is available online [60] to ensure full reproducibility.

B. Performance measures

To assess performance we use the framework made available online [39] by Vivone et al., and described in [25]. Accordingly, we report results in terms of multiple performance measures. In fact, no single measure can be considered as a fully reliable indicator of pansharpening quality, and it is therefore good practice to take into account different perspectives. In particular, it is advisable to consider both full-reference (low-resolution) and full-resolution (no-reference) measures.

Following the Wald protocol [42], [10], [25], full-reference measures are computed on the reduced resolution dataset, so as to use the original MS data as reference. Therefore, they can measure pansharpening accuracy objectively. On the down side, the reduced-resolution data are obtained through a downgrading procedure which may introduce a bias in the accuracy evaluation. A method that performs well on erroneously downgraded data may turn out to work poorly at full resolution.

The choice of the low-pass anti-aliasing filter which precedes decimation is therefore a crucial issue of this approach. In the original paper by Wald [42] it is left as an open concern. Here, we adopt the solution proposed by Aiazzi et al. [16], and implemented in the Open Remote Sensig repository [39], [25], which uses low-pass filters matched with the MTFs of the different channels of the sensor, considering that these responses may be significantly different from one MS band to another and also be rather different from the PAN.

Here, we use the following widespread full-reference measures, referring to the original papers for their thorough description:

- **SAM**: Spectral Angle Mapper [61];
- **ERGAS**: Global adimens. relative synthesis error [62];
- **Q**: Average universal image Quality index [63];
- **Q4 / Q8**: 4 / 8-band extension of Q [64].

Full-resolution measures, instead, work on the original data, thus avoiding any biases introduced by the downgrading procedure. In particular, we consider here the QNR and its components, referring again to the original paper for all details:

- **QNR**: Quality with No-Reference index [65];
- **Dλ**: Spectral component of QNR;
- **DS**: Spatial component of QNR.

Their major drawback is, obviously, the absence of reference data at full resolution, which undermines the measures’ objectivity. In addition, these measures rely on the upsampled MS image as a guide to compute intermediate quantities, thereby introducing their own biases. In particular, for the EXP method, performing only MS upsampling, the Dλ measure vanishes altogether, leading to a very high QNR despite a clear loss of resolution and very poor results on reference-based measures. More in general, DS is biased for any method approaching EXP.

In summary, the absence of true reference data makes assessment a challenging task. Accordingly, a good pansharpening quality may be claimed when all measures, with and without reference, are good, while results that change very much across measures suggest biases and poor quality.

C. Datasets and training

To assess performance in a wide variety of operating conditions we experimented with images acquired by several multiresolution sensors, Ikonos, GeoEye-1, WorldView-2 and WorldView-3, with scenes concerning both rural and urban areas, taken in different countries. Tab. II reports the list of all test datasets. Accordingly, the spatial resolution measured on the PAN component varies from 0.82m (Ikonos) to 0.31m (WorldView-3), while 4 (Ikonos, GeoEye-1) to 8 (WorldView) bands are available, covering the visible and near-infrared regions of the spectrum.

Our main goal, however, is to study performance as a function of the mismatch between training and test data. Under this point of view, we classify operating conditions into favourable, typical, and challenging. Favourable conditions occur when training and test sets, although separated, are taken from the same image, and hence share all major statistical features. It is worth underlining that, due to lack of data, this is quite a common case, and the only one explored in our previous work [34]. More typically, training and test data are expected to be unrelated. For example, a network trained on the Caserta-WorldView-2 scene may be used for the pansharpening of the WorldView-2 image of Stockholm. Finally, one may try to use a network trained on a given sensor...
to process images acquired from a different one. For example, the network trained on the Caserta-WorldView-2 scene, may be used to pansharpen a WorldView-3 image. In Table III together with each test dataset, we report the corresponding training set, with combinations covering all operating conditions of interest. Note that only the Caserta datasets were used for training. In particular, for each sensor, a training/validation set was generated, comprising 14400/7200 tiles of $33 \times 33$ pixels, collected in mini-batches of 128 elements for an efficient implementation of the stochastic gradient descent algorithm. The training procedure is the same already used in [34] where additional details can be found. Eventually, the trained nets are tested on a number of $1280 \times 1280$ clips (PAN resolution), disjoint from the training set.

### D. Discussion of results

We start our analysis from the favourable cases of Tables III, IV and V (discard the last two rows of Tables III, IV for the time being), where both training set and test set are drawn from the same image of Caserta, acquired by the Ikonos, GeoEye-1, and WorldView-2 sensors, respectively. Results for conventional techniques are grouped in the upper part of the table, with the best result for each indicator shown in boldface blue. Then, we have a single line for the baseline method, PNN with augmented input, implemented in Theano. The following three rows show results for the proposed variations, with L1 loss, residual learning, and fine tuning, with the best result in boldface red. For these cases, it was already shown, in [34], that PNN improves significantly and almost uniformly over all measures with respect to reference methods. Hence, we focus on the further modifications introduced in this work. With respect to the baseline, L1 loss and residual learning guarantee small but consistent improvements, uniformly over all measures. As for the fine tuning step, it improves all full-reference measures but not all no-reference ones, especially $D_\lambda$, probably due to the bias introduced in its computation. In any case, the variant with fine tuning is almost always the best CNN-based solution, and best overall, with some exceptions only for no-reference measures. The limited improvement w.r.t. the baseline is explained by the very good results obtained by the latter in these favourable conditions. It is worth reminding that PNN performs already much better than all conventional methods, and the gap has now grown even wider. For the sake of brevity, we do not show visual results for these relatively less interesting cases.

Let us now consider the typical case, when training and test data are acquired with the same sensor but come from different scenes. To this end we report in Tables VII and VIII results for WorldView-2 data, with network trained on the Caserta image and used on the Stockholm and Washington images. In this case results are much more controversial. PNN keeps providing good results, but not always superior to conventional methods, especially MTF-GLP-HPM on full-reference measures, and PRACS on no-reference measures. On Stockholm, in particular, a large $D_\lambda$ value is observed, testifying of a poor spectral fidelity, which impacts on the overall QNR, 4 percent points worse than PRACS. The adoption of L1 loss and residual learning provides mixed effects, mostly minor losses at low resolution and minor gains at full resolution. On the contrary, fine tuning has a strong impact on performance, leading this version of PNN to achieve the best results almost uniformly on all measures and for both images. For full reference measures, in particular, the fine-tuning version provides a huge gain with respect to both the baseline and the best conventional method, On Stockholm, for example, SAM lowers from 7.45 to 4.82 and ERGAS from 5.58 to 3.72. In terms of QNR it approaches very closely the best conventional methods, while the best performance is given by naive interpolation (EXP), suggesting to take this indicator with some care.

Visual inspection helps explaining this behavior. In Fig. 7 (a) and (b), with reference only to the proposed methods, we show some sample results for Stockholm and Washington, respectively. Here and in the following figures, we show in the first row the PAN image together with full-resolution pansharpening results, so as to appreciate spatial accuracy. The second row, instead, shows the MS image followed by reduced-resolution pansharpening results, corresponding to full-reference measures, and providing information on spectral fidelity. Then, in the third row, we show the difference between reduced-resolution pansharpening and MS, to better appreciate errors. In each figure, all images are subject to the same histogram stretch to improve visibility, except for the difference images that are further enhanced. The Stockholm image explains very clearly the relatively poor performance of PNN. Because of the mismatch between training and test
### TABLE III
**Performance indicators on the Caserta-Ikonos dataset**

|                  | FULL-REFERENCE |                | NO-REFERENCE |                |
|------------------|----------------|----------------|--------------|----------------|
|                  | Q4  | Q | SAM | ERGAS | \(D_\lambda\) | \(D_S\) | QNR |
| EXP              | 0.4476 | 0.7050 | 3.0090 | 2.8386 | 0             | 0.0438 | 0.9562 |
| PRACS            | 0.6397 | 0.8021 | 2.9938 | 2.3597 | 0.0493        | 0.1148 | 0.8424 |
| Indusion         | 0.5928 | 0.7660 | 3.2800 | 2.7961 | 0.1264        | 0.1619 | 0.7340 |
| AWLP             | 0.7143 | 0.8389 | 2.8426 | 2.1126 | 0.1384        | 0.1955 | 0.6951 |
| ATWT-M3          | 0.5579 | 0.7249 | 3.5807 | 3.0327 | 0.1244        | 0.1452 | 0.7490 |
| MTF-GLP-HPM      | 0.7178 | 0.8422 | 2.8820 | 2.0550 | 0.1524        | 0.2186 | 0.6646 |
| BDSD             | 0.7199 | **0.8576** | 2.9147 | **1.9852** | **0.0395** | **0.0884** | **0.8761** |
| C-BDSD           | **0.7204** | 0.8569 | 2.9101 | 2.0553 | 0.0710        | 0.1218 | 0.8173 |
| Baseline (Theano)| 0.7689 | 0.9056 | 2.1999 | 1.5885 | 0.0542        | 0.0698 | 0.8800 |
| L1               | 0.7811 | 0.9124 | 2.0745 | 1.5077 | 0.0285        | 0.0676 | 0.9062 |
| L1-RL            | 0.7868 | 0.9165 | 2.0218 | 1.4705 | 0.0220        | 0.0635 | 0.9158 |
| L1-RL-FT         | **0.7969** | **0.9230** | **1.9390** | **1.4013** | **0.0207** | 0.0641 | **0.9166** |
| L1-RL (cross-sensor) | 0.6742 | 0.8321 | 3.0163 | 2.3345 | 0.0249        | 0.0708 | 0.8783 |
| L1-RL (cross-sensor)-FT | 0.7753 | 0.9080 | 2.1622 | 1.5323 | 0.0249        | 0.0687 | 0.9082 |

### TABLE IV
**Performance indicators on the Caserta-GeoEye-1 dataset**

|                  | FULL-REFERENCE |                | NO-REFERENCE |                |
|------------------|----------------|----------------|--------------|----------------|
|                  | Q8  | Q | SAM | ERGAS | \(D_\lambda\) | \(D_S\) | QNR |
| EXP              | 0.5178 | 0.8177 | 2.3393 | 1.8160 | 0             | 0.0832 | 0.9168 |
| PRACS            | 0.6995 | 0.8568 | 3.2364 | 2.4296 | **0.0470** | **0.0877** | **0.8698** |
| Indusion         | 0.5743 | 0.7771 | 3.5361 | 3.5480 | 0.1270        | 0.1262 | 0.7651 |
| AWLP             | 0.7175 | 0.8615 | 3.6297 | 2.6134 | 0.1247        | 0.1521 | 0.7436 |
| ATWT-M3          | 0.6068 | 0.7907 | 3.5546 | 3.0729 | 0.0712        | 0.0710 | 0.8363 |
| MTF-GLP-HPM      | 0.7359 | 0.8718 | **3.2205** | 5.0344 | 0.1526        | 0.1815 | 0.6956 |
| BDSD             | **0.7399** | **0.8832** | 3.3384 | **2.2342** | 0.0490        | 0.0994 | 0.8572 |
| C-BDSD           | 0.7391 | 0.8784 | 3.4817 | 2.4370 | 0.0832        | 0.1342 | 0.7953 |
| Baseline (Theano)| 0.8150 | 0.9428 | 2.1011 | 1.4965 | 0.0377        | 0.0615 | 0.9034 |
| L1               | 0.8219 | 0.9456 | 2.0695 | 1.4595 | **0.0250** | **0.0584** | **0.9181** |
| L1-RL            | 0.8203 | 0.9461 | 2.0579 | 1.4798 | 0.0296        | **0.0485** | **0.9235** |
| L1-RL-FT         | **0.8290** | **0.9509** | **2.0070** | **1.4137** | **0.0273** | 0.0523 | **0.9220** |
| L1-RL (cross-sensor) | 0.6544 | 0.8369 | 3.4486 | 2.6594 | 0.0478        | 0.1077 | 0.8498 |
| L1-RL (cross-sensor)-FT | 0.8054 | 0.9402 | 2.2739 | 1.5820 | 0.0286        | 0.0585 | 0.9148 |

### TABLE V
**Performance indicators on the Caserta-WorldView-2 dataset**

|                  | FULL-REFERENCE |                | NO-REFERENCE |                |
|------------------|----------------|----------------|--------------|----------------|
|                  | Q8  | Q | SAM | ERGAS | \(D_\lambda\) | \(D_S\) | QNR |
| EXP              | 0.4963 | 0.7125 | 4.9443 | 5.7764 | 0             | 0.0653 | 0.9347 |
| PRACS            | 0.7908 | 0.8789 | 3.6995 | 2.4102 | **0.0234** | 0.0734 | 0.9050 |
| Indusion         | 0.6928 | 0.8373 | 3.7261 | 3.2022 | 0.0552        | 0.0649 | 0.8839 |
| AWLP             | 0.8127 | 0.9043 | **3.4182** | 2.2560 | 0.0665        | 0.0849 | 0.8549 |
| ATWT-M3          | 0.7039 | 0.8186 | 4.0655 | 3.1609 | 0.0675        | 0.0748 | 0.8628 |
| MTF-GLP-HPM      | **0.8242** | **0.9083** | 3.4497 | **2.0918** | 0.0755        | 0.0953 | 0.8373 |
| BDSD             | 0.8110 | 0.9052 | 3.7449 | 2.2644 | 0.0483        | **0.0382** | **0.9156** |
| C-BDSD           | 0.8004 | 0.8948 | 3.9891 | 2.6363 | 0.0251        | 0.0458 | **0.9304** |
| Baseline (Theano)| 0.8681 | 0.9544 | 2.3419 | 1.4341 | 0.0195        | 0.0464 | 0.9350 |
| L1               | 0.8709 | 0.9562 | 2.3056 | 1.4221 | **0.0164** | **0.0489** | **0.9355** |
| L1-RL            | 0.8716 | 0.9581 | 2.2927 | 1.4385 | 0.0238        | **0.0404** | **0.9367** |
| L1-RL-FT         | **0.8771** | **0.9615** | **2.1913** | **1.3685** | 0.0216        | 0.0428 | 0.9366 |
data, pansharpened images are affected by a large spectral error, both at low and high resolution, with a dominant green hue. The problem is almost completely corrected by resorting to residual learning, which works on differences and, hence, tends to reduce biases. However, there is still a clear loss of spatial resolution in the low-resolution image, as testified by structures in the error image. Fine tuning solves this latter problem as well, providing satisfactory results in terms of both spectral and spatial resolution. The error image shows neither dominant hues (spectral errors) nor marked structures (spatial errors). Similar considerations apply to the Washington image, even though in this case the spectral bias is much smaller, due to a better alignment of training and test data.

In Fig. 7 the visual comparison is against reference methods. The detail on top is particularly interesting because it includes an industrial plant with a large roof writing, the ideal image for visual appreciation of spatial resolution. Let us first consider the EXP image (simple interpolation): despite the very strong blurring, it has the best no-reference measures, further testifying on the need to consider multiple quality indicators, together with visual inspection. At full resolution (first row) the proposed method is among the best, but some reference methods work equally well, notably MTF-GLP-HTM, Indusion, AWLP, PRACS and ATWT-M3, instead, exhibit over-smoothing, while C-BDSD and ATWT-M3 are affected by spectral distortion. At reduced resolution, the proposed method seem clearly superior to all references, as predicted by objective measures and also confirmed by the error images in the third row. Again, similar considerations with minor differences apply to the Washington image (bottom detail).

Finally, let us consider the most challenging case of cross-sensor pansharpening. Going back to Tables III and IV in the last two rows we report results obtained with our best proposed method, L1-RL, using a cross-sensor network with and without fine tuning. In detail, for Tab. III concerning the Caserta-Ikonos image, the network is trained on Caserta-GeoEye-1 data, and viceversa for Tab. IV. In the absence of fine tuning (penultimate row) there is a large loss of performance w.r.t same-sensor training, and even w.r.t. conventional methods, at least for full-reference measures. This gap, however, is almost completely recovered through fine tuning, achieving a performance which is only slightly inferior to the overall best. It is worth reminding that this result is obtained with a negligible computational effort, while training a net from scratch would require many hours even with suitable GPUs.

Tab. VIII shows results for the Adelaide image, acquired with the WorldView-3 sensor, using the network trained on WorldView-2 data. In this case there is no same-sensor comparison, we just used the available WorldView-2 net. Again, after fine tuning, the proposed method works much better that conventional methods for full-reference measures and comparable to the best of them, C-BDSD, for full-resolution measures. The visual inspection of Fig. 9 however, makes clear that C-BDSD is affected by a large spectral error, while other methods cause some loss of resolution. Overall, thanks
Fig. 8. Output of proposed and reference methods on clips from the Stockholm-WorldView-2 (top) and Washington-WorldView-2 (bottom) images.
TABLE VI

| PERFORMANCE INDICATORS ON THE STOCKHOLM-WORLDVIEW-2 DATASET |
|-------------------------------------------------------------|
|               | FULL-REFERENCE | NO-REFERENCE |
|               | Q8  | Q   | SAM | ERGAS | D_\lambda | D_\delta | QNR |
| (desired value) |     |     |     |       |   (0)     |   (0)    |     |
| EXP            | 0.3693 | 0.5916 | 8.4012 | 9.9350 | 0.0522 | 0.9478 |
| PRACS          | 0.6964 | 0.7910 | 7.9066 | 7.0412 | **0.0126** | 0.9935 | **0.8952** |
| Indusion       | 0.6284 | 0.7674 | 8.7207 | 7.6498 | 0.0589 | 0.1084 | 0.8403 |
| AWLP           | 0.7486 | 0.8388 | 7.3534 | 6.0312 | 0.0624 | 0.1166 | 0.8292 |
| ATWT-M3        | 0.6339 | 0.7301 | 8.2871 | 8.2369 | 0.0849 | 0.1311 | 0.7964 |
| MTF-GLP-HPM    | **0.7576** | **0.8489** | **7.1035** | **5.6989** | 0.0749 | 0.1316 | 0.8047 |
| BDS            | 0.7427 | 0.8457 | 8.3054 | 6.0552 | 0.0820 | 0.1544 | 0.7762 |
| C-BDS          | 0.7428 | 0.8482 | 8.1303 | 6.5450 | 0.0543 | **0.0634** | 0.8861 |
| Baseline (Theano) | 0.7306 | 0.8539 | 7.4533 | 5.8517 | 0.1064 | 0.0444 | 0.8539 |
| L1             | 0.6490 | 0.8232 | 8.0037 | 6.0525 | 0.0724 | 0.0503 | 0.8810 |
| L1-RL          | 0.6672 | 0.8998 | 6.9955 | 6.2996 | 0.0496 | **0.0438** | **0.9087** |
| L1-RL-FT       | **0.8332** | **0.9141** | 4.8256 | 3.7237 | 0.0427 | 0.0713 | 0.8896 |

TABLE VII

| PERFORMANCE INDICATORS ON THE WASHINGTON-WORLDVIEW-2 DATASET |
|-------------------------------------------------------------|
|               | FULL-REFERENCE | NO-REFERENCE |
|               | Q8  | Q   | SAM | ERGAS | D_\lambda | D_\delta | QNR |
| (desired value) |     |     |     |       |   (0)     |   (0)    |     |
| EXP            | 0.3833 | 0.6153 | 7.4957 | 7.1793 | 0.0578 | 0.9422 |
| PRACS          | 0.6669 | 0.7744 | 7.4039 | 5.3591 | **0.0124** | 0.9950 | 0.8938 |
| Indusion       | 0.6618 | 0.7826 | 7.5296 | 5.5846 | 0.0704 | 0.1090 | 0.8287 |
| AWLP           | 0.7527 | 0.8418 | **7.0800** | 4.5701 | 0.0744 | 0.1276 | 0.8080 |
| ATWT-M3        | 0.6023 | 0.7163 | 8.2031 | 6.1242 | 0.0666 | 0.1348 | 0.8080 |
| MTF-GLP-HPM    | **0.7667** | **0.8490** | **7.1273** | **4.4506** | 0.0831 | 0.1422 | 0.7869 |
| BDS            | 0.7505 | 0.8436 | 8.0870 | 6.4441 | 0.0782 | 0.1100 | 0.8207 |
| C-BDS          | 0.7548 | 0.8437 | 8.2856 | 4.9512 | 0.0471 | **0.0424** | **0.9127** |
| Baseline (Theano) | 0.7891 | 0.8850 | 5.4310 | 3.5606 | 0.0444 | 0.0634 | 0.8953 |
| L1             | 0.7798 | 0.8822 | 5.6451 | 3.5928 | 0.0426 | 0.0636 | 0.8966 |
| L1-RL          | 0.7801 | 0.8837 | 5.5199 | 3.6605 | 0.0441 | 0.0507 | 0.9074 |
| L1-RL-FT       | **0.8572** | **0.9291** | **4.6544** | **2.7935** | **0.0410** | **0.0498** | **0.9108** |

VI. Conclusions

We started from our recently proposed CNN-based pansharpening method, featuring already a state-of-the-art performance, and explored a number of architectural and operating variations to improve both quality and robustness. When the training set is well matched to the test data, residual learning and L1 loss ensure some limited improvements, together with a significant speed-up in training. The most interesting results, however, are observed in the presence of training-test mismatch, quite common in remote sensing due to the scarcity of free data. In this case, target adaptation, obtained through a fine tuning pass, provides a very significant performance gain, coming with negligible computational cost, and with no active user involvement.

Full-resolution quality remains an open issue. Indeed, while the performance is fully satisfactory on subsampled data, there is still room for improvements at the highest resolution. Besides a better modeling of the MTF, and a better compensation of atmospheric effects, a major impact may come from the design of more reliable no-reference measures. These would enable training and fine tuning based on a task-specific loss function, with a sure impact on performance.

REFERENCES

[1] V. Shettigara, “A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set,” *Photogrammetric Engineering & Remote Sensing*, vol. 58, no. 5, pp. 561–567, 1992.

[2] T.-M. Tu, S.-C. Su, H.-C. Shyu, and P. S. Huang, “A new look at ihs-like image fusion methods,” *Information Fusion*, vol. 2, no. 3, pp. 177 – 186, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253501000367.

[3] T.-M. Tu, P. S. Huang, C.-L. Hung, and C.-P. Chang, “A fast intensity-hue-saturation fusion technique with spectral adjustment for ikonos imagery,” *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, pp. 309–312, 2004.

[4] J. Chavez and A. Kwarteng, “Extracting spectral contrast in landsat the-"

[5] A. R. Gillespie, A. B. Kahle, and R. E. Walker, “Color enhancement of highly correlated images. II. Channel ratio and “chromaticity” transformation techniques,” *Remote Sensing of Environment*, vol. 22, no. 3, pp. 339 – 348, 1989.

[6] C. Laben, and B. Brower, “Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening.” *U.S. Patent 6011875*, 2000., 2000.
Fig. 9. Output of proposed and reference methods on a clip from the Adelaide-WorldView-3 image.

| TABLE VIII | PERFORMANCE INDICATORS ON THE WV-3/Adelaide DATASET |
|------------|-----------------------------------------------------|
|            | FULL-REFERENCE                                    | NO-REFERENCE                                    |
|            | Q8  | Q  | SAM  | ERGAS | D_0 | D_2 | QNR |
| EXP        | 0.3973 | 0.6506 | 7.4100 | 7.7729 | 0 | 0.0716 | 0.9284 |
| PRACS      | 0.7158 | 0.8311 | 7.3511 | 5.2708 | 0.0169 | 0.1018 | 0.8832 |
| Indusion   | 0.6579 | 0.8026 | 7.5855 | 6.0244 | 0.0564 | 0.1170 | 0.8335 |
| AWLP       | 0.7628 | 0.8632 | 7.3040 | 4.8743 | 0.0578 | 0.1069 | 0.8417 |
| ATWT-M3    | 0.6206 | 0.7614 | 7.9917 | 6.3834 | 0.0765 | 0.1052 | 0.8265 |
| MTF-GLP-HPM| 0.7780 | 0.8758 | 7.0625 | 5.1554 | 0.0795 | 0.1291 | 0.8019 |
| BDSD       | 0.7876 | 0.8828 | 7.4218 | 4.5010 | 0.0441 | 0.0764 | 0.8830 |
| C-BDSD     | 0.7882 | 0.8821 | 7.5475 | 5.4875 | 0.0297 | 0.0532 | 0.9188 |
| Baseline (cross-sensor) | 0.7430 | 0.8639 | 8.1817 | 4.9629 | 0.0702 | 0.0641 | 0.8701 |
| L1 (cross-sensor) | 0.6965 | 0.8375 | 9.5438 | 5.9146 | 0.0740 | 0.0649 | 0.8659 |
| L1-RL (cross-sensor) | 0.6943 | 0.8360 | 7.7872 | 5.3905 | 0.0894 | 0.0473 | 0.8676 |
| L1-RL (cross-sensor)-FT | **0.8546** | **0.9376** | **5.0932** | **3.2556** | **0.0393** | **0.0474** | **0.9151** |

[7] 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.

[8] 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, no. 1, pp. 295–309, Jan 2011.

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

[10] T. Ranchin and L. Wald, “Fusion of high spatial and spectral resolution images: the ARSIS concept and its implementation,” *Photogrammetric engineering and remote sensing*, vol. 66, no. 1, pp. 49–61, 2000.

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

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

[13] M. Khan, J. Chanussot, L. Condat, and A. Montanvert, “Indusion: Fusion of multispectral and panchromatic images using the induction scaling technique,” *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 1, pp. 98–102, Jan 2008.

[14] 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.

[15] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, “An MTF-based spectral distortion minimizing model for pan-sharpening of very
high resolution multispectral images of urban areas,” in GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, May 2003, pp. 90–94.

[16] ———, “Mtf-tailored multiscale fusion of high-resolution ms and pan imagery,” Photogrammetric Engineering & Remote Sensing, vol. 72, no. 8, pp. 951–958, 2006.

[17] J. Lee and C. Lee, “Fast and efficient panchromatic sharpening,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 1, pp. 155–163, Jan 2010.

[18] R. Restaino, M. D. Mura, G. Vivone, and J. Chanussot, “Context-adaptive pansharpening based on image segmentation,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 2, pp. 753–766, Feb 2017.

[19] V. P. Shah, N. H. Younan, and R. L. King, “An efficient pan-sharpening method via a combined adaptive pca approach and contourlets,” IEEE Transactions on Geoscience and Remote Sensing, vol. 46, no. 5, pp. 1323–1335, May 2008.

[20] 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, June 2008.

[21] F. Palsson, J. Sveinsson, and M. Ulfarsson, “A new pansharpening algorithm based on total variation,” Geoscience and Remote Sensing Letters, IEEE, vol. 11, no. 1, pp. 318–322, Jan 2014.

[22] A. Garzelli, “Pansharpening of multispectral images based on nonlocal parameter optimization,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 4, pp. 2094–2107, April 2015.

[23] J. Duran, A. Buades, B. Coll, C. Sbert, and G. Blanchet, “A survey of pansharpening methods with a new band-decoupled variational model,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 125, pp. 78 – 105, 2017.

[24] S. Zhong, Y. Zhang, Y. Chen, and D. Wu, “Combining component substitution and multisolution analysis: A novel generalized bsd pansharpening algorithm,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 6, pp. 2867–2875, June 2017.

[25] G. Vivone, L. Alparone, J. Chanussot, M. D. Mura, A. Garzelli, G. A. 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.

[26] B. Aiazzi, L. Alparone, S. Baronti, R. Carl, A. Garzelli, and L. Santurri, “Sensitivity of pansharpening methods to temporal and instrumental changes between multispectral and panchromatic data sets,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 1, pp. 308–319, Jan 2017.

[27] L. Alparone, S. Baronti, B. Aiazzi, and A. Garzelli, “Spatial methods for multispectral pansharpening: Multiresolution analysis demystified,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 5, pp. 2563–2576, May 2016.

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

[29] S. Li, H. Yin, and L. Fang, “Remote sensing image fusion via sparse representations over learned dictionaries,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 9, pp. 4779–4789, Sept 2013.

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

[31] M. Cheng, C. Wang, and J. Li, “Sparse representation based pansharpening using trained dictionary,” IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 1, pp. 293–297, 2014.

[32] X. Z. Zhu, C. Grohfeld, and R. Bamler, “Exploiting joint sparsity for pansharpening: The j-sparsefi algorithm,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3181–3196, July 2017.

[33] S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” arXiv preprint arXiv:1408.5093, 2014.

[34] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3171–3185, July 2017.

[35] M. M. Khan, L. Alparone, and J. Chanussot, “Pansharpening quality assessment using the modulation transfer functions of instruments,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 5, pp. 3880–3891, Nov 2009, 2013.

[36] B. Aiazzi, L. Alparone, S. Baronti, R. Carl, A. Garzelli, and L. Santurri, “Full scale assessment of pansharpening methods and data products,” vol. 9244, 2014, pp. 9244 – 9244 – 12.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 770–778.

[38] I. Sutskever, J. Martens, G. E. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” JMLR (3), vol. 28, pp. 1139–1147, 2013.
[64] L. Alparone, S. Baronti, A. Garzelli, and F. Nencini, “A global quality measurement of pan-sharpened multispectral imagery,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 1, no. 4, pp. 313–317, Oct 2004.

[65] 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, February 2008.