Target-adaptive CNN-based pansharpening

Giuseppe Scarpa, Sergio Vitale, and Davide Cozzolino

***Abstract*—We recently proposed a convolutional neural network (CNN) 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 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].

G. Scarpa and D. Cozzolino are with the Department of Electrical Engineering and Information Technology, University Federico II, Naples, Italy, e-mail: {firstname.lastname}@unina.it. S. Vitale is with the Engineering Department, University Parthenope, Naples, Italy, e-mail: sergio.vitale@uniparthenope.it.

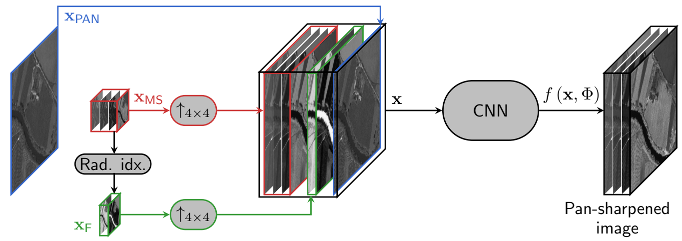


Fig. 1. General workflow of the PNN algorithm [41].

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 [20], [21], [22], [23], [24], [25]. or variational [26], [27] that get state of the art results. However this CS-MRA dichotomy is useful to understand the behaviour of any method falling in these categories as highlighted in [28], [29]. 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 [30]. 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 [29].

In the last few 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 [31], [32], [33], [34], [35], [36]. 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 [37], [38], [39], and several papers have been proposed to address pansharpening [40], [41], [42], [43], [44], [45]. In particular, our CNN-based solution [41], inspired by work on super-resolution [46], provided a significant performance improvement with respect to the previous state-of-the-art.

In this work we start from the baseline solution of [41] 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 [47], under a wide range of quality measures.

In the rest of the paper we describe the baseline method (Section II), the proposed architectural improvements (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 [41], [48], inspired to previous work on super-resolution [46], 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 $\mathbf{x}_{\rm PAN}$ (blue), the multispectral component $\mathbf{x}_{\rm MS}$ (red) up-sampled via polynomial interpolation, and a few radiometric indices $\mathbf{x}_{\rm F}$ (green) extracted from the MS component and interpolated as well. The latter component, comprising some nonlinear combinations of MS spectral bands, has proven experimentally to improve the network performance.

The CNN is composed by three convolutional layers with nonlinear 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, $\mathbf{w}$, accounting for the weights, and a M-vector, $\mathbf{b}$, for the biases. For layer l, with input $\mathbf{x}^{(l)}$, the filter output is computed as

$$\mathbf{z}^{(l)} = \mathbf{w}^{(l)} * \mathbf{x}^{(l)} + \mathbf{b}^{(l)},$$

where the m-th component can be expressed as

$$\mathbf{z}^{(l)}(m,\cdot,\cdot) = \sum_{n=1}^{N} \mathbf{w}^{(l)}(m,n,\cdot,\cdot) * \mathbf{x}^{(l)}(n,\cdot,\cdot) + \mathbf{b}^{(l)}(m)$$

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

$$\mathbf{y}^{(l)} = f_l(\mathbf{x}^{(l)}, \Phi_l) = \max(0, \mathbf{z}^{(l)}) \qquad (1)$$

where $\Phi_l \triangleq \left(\mathbf{w}^{(l)}, \mathbf{b}^{(l)}\right)$. The choice of ReLU nonlinearities is motivated experimentally [49] by the good convergence properties they guarantee. Notice that, as neither stride nor pooling are used, each layer preserves the input resolution and hence $\mathbf{x}^{(l)}$ and $\mathbf{y}^{(l)}$ have the same spatial size.

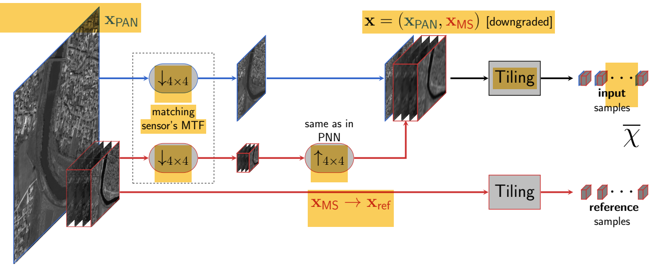


Fig. 2. Generation of a training dataset through Wald's protocol.

A. Learning

Let $\mathbf{x} = (\mathbf{x}_{\rm PAN}, \mathbf{x}_{\rm MS}, \mathbf{x}_{\rm F})$ denote the composite CNN input stack and $f(\mathbf{x}, \Phi)$ 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 [50], 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. Fig. 2 shows how Wald's protocol is used to create training examples. Before downsampling, a low-pass filter is applied to reduce aliasing. To match the sensor properties, in [16] it is proposed to use an approximation of the sensor MTF. This scheme generalizes readily to the case in which additional low-resolution input bands are considered, like the MS radiometric indices.

Given a training set $\overline{\chi} = \{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_Q\}$, generated by Wald's protocol with any MTF modeling, comprising Q input-output image pairs $\overline{\mathbf{x}} \triangleq (\mathbf{x}, \mathbf{x}_{\rm ref})$, the objective of the training phase is to find

$$\Phi = \arg\min_{\Phi} J\left(\overline{\chi}, \Phi\right) \triangleq \arg\min_{\Phi} \frac{1}{Q} \sum_{\overline{\mathbf{x}} \in \overline{\chi}} L(\overline{\mathbf{x}}, \Phi)$$

where $L(\overline{\mathbf{x}}, \Phi)$ is a suitable loss function.

In [41] the mean square error (MSE) was used as loss function[1]

$$L(\overline{\mathbf{x}}, \Phi) \propto ||f(\mathbf{x}, \Phi) - \mathbf{x}_{\rm ref}||_2^2$$

and the minimization was carried out by stochastic gradient descent (SGD) with momentum [51]. In particular, the training set was partitioned in P batches, $\{B_1, \ldots, B_P\}$, with $\bigcup_j B_j = \overline{\chi}$, and at each iteration a batch was sampled and used to estimate the gradient and update parameters as

$$\begin{aligned} \nu^{(n+1)} &\leftarrow \mu\nu^{(n)} + \alpha \nabla_{\Phi} J\left(B_{j_n}, \Phi^{(n)}\right) \\ \Phi^{(n+1)} &\leftarrow \Phi^{(n)} - \nu^{(n+1)} \end{aligned}$$

Training efficiency and accuracy depend heavily on the algorithm hyperparameters, learning rate, α, momentum, μ, and

[1] To account for border effects, the norm is computed on cropped versions of $f(\mathbf{x}, \Phi)$ and $\mathbf{x}_{\rm ref}$.

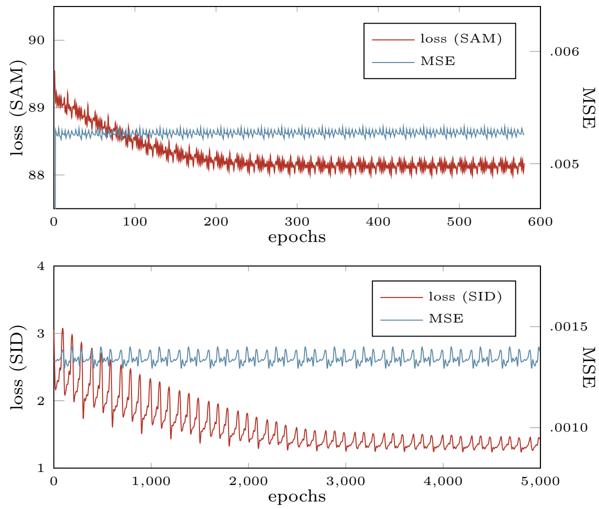


Fig. 3. Loss and MSE during training using SAM (top) or SID (bottom) as loss.

velocity, ν, the most critical being the learning rate, which can cause instability when too large or slow convergence when too small. In [41], after extensive experiments, the learning rate was set to 10^{-4} for Φ_1 and Φ_2 and to 10^{-5} for Φ_3, which ensured convergence in about 10^6 iterations.

III. Improving CNN-based pansharpening

Although the PNN architecture relies on solid conceptual foundations, following a well-motivated path from dictionary-based super-resolution [52], to its CNN-based counterpart [46], 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 [28], [41], but some of them are quite complicated, hence time-consuming, and may happen to be unstable. As an example one may think to use typical measures employed for quality assessment or discrimination in multispectral or hyperspectral image analysis, like the spectral angle mapper (SAM) [53] or the spectral information divergence (SID) [54]. Unless some form of regularization is considered, these choices are unsuited for optimization as they are insensitive to intensity scaling and because they induce highly complex targets, full of local minima, where one can easily get trapped during training. This can also be observed experimentally with the help of Fig. 3, where we plot the loss (SAM or SID) along with

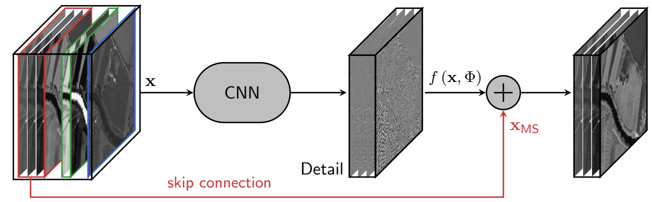


Fig. 4. Proposed residual-based architecture. Preprocessing not shown.

the a common measure of distortion, the mean square error (MSE), as function of the traininig iterations. As it can be seen, the losses do not reach reasonable levels of accuracy, getting trapped in some local minima. Infact, typical values should be a few degrees, for SAM, and some order of magnitude lesser than the unit, in case of SID. Furthermore, as we will see later, for our dataset we expect the MSE to be one order of magnitude smaller than the values registered in Fig. 3. More in general, the MSE do not follow the trend of the loss, which is an indirect confirm of the scaling insensity of the selected distances.

Motivated by the above observations we decided to restrict our study to Ln norms, as the reduction of the training time is a primary goal in this work, leaving to future research the aim of improving this choice further. In [41] we used the L2 loss, as in [46], 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 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 [55]. 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 [41]. 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 [41], 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.4, where the preprocessing is omitted for the sake of simplicity. The core CNN is trained to generate only the

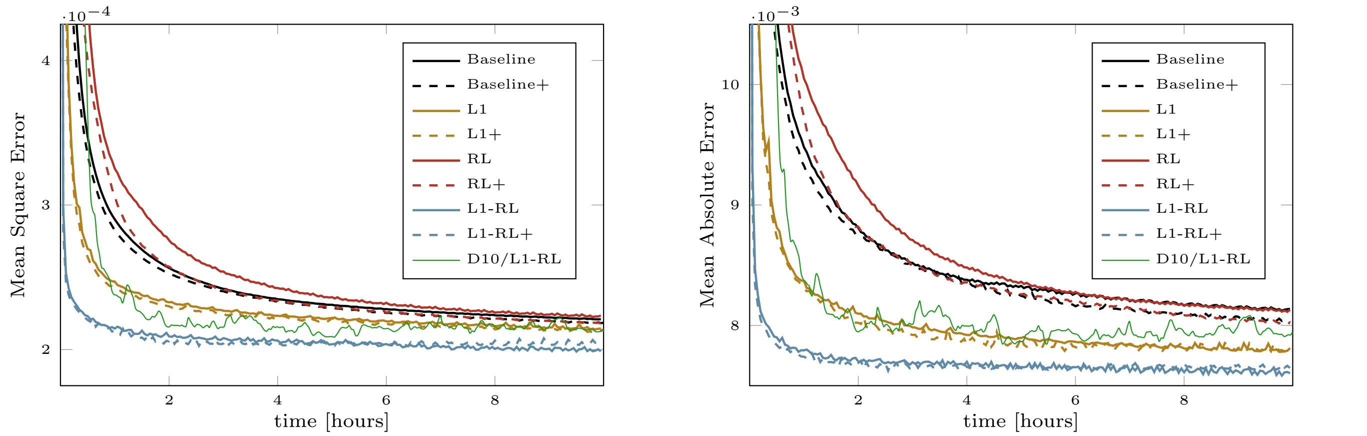


Fig. 5. Mean square error (MSE, left) and mean absolute error (MAE, right) of the proposed methods on the GeoEye-1 validation set. Dashed lines (+ in the legend) indicate input augmentation with radiometric indices. Accordingly, baseline+ coincides with PNN [41]. L1 indicates use of L1-loss in place of L2-loss for training. RL indicates use of residual learning. D10 marks the best deep architecture (10 layers) selected by preliminary tests.

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, $\mathbf{x}_{\mathrm{MS}}$, made available through a skip connection, to the network output, $f(\mathbf{x}, \Phi)$. The loss is then computed as

$$\begin{aligned} L(\overline{\mathbf{x}}, \Phi) &\propto ||(f(\mathbf{x}, \Phi) + \mathbf{x}_{\mathrm{MS}}) - \mathbf{x}_{\mathrm{ref}}||_2^2 \\ &\propto ||f(\mathbf{x}, \Phi) - \Delta\mathbf{x}_{\mathrm{ref}}||_2^2 \end{aligned}$$

with the residual reference defined as $\Delta\mathbf{x}_{\mathrm{ref}} \triangleq \mathbf{x}_{\mathrm{ref}} - \mathbf{x}_{\mathrm{MS}}$ and, accordingly, the training samples redifined as $\overline{\mathbf{x}} \triangleq (\mathbf{x}, \Delta\mathbf{x}_{\mathrm{ref}})$. 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 $\mathbf{x}_{\mathrm{MS}}$.

If we also replace the L2 norm with L1 norm as suggested in the previous subsection, the loss becomes eventually

$$L(\overline{\mathbf{x}}, \Phi) \propto ||f(\mathbf{x}, \Phi) - \Delta\mathbf{x}_{\mathrm{ref}}||_1$$

Note that residual learning is not a new idea. In [56], [57] 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 [58], [59], and used successfully in various applicative fields, e.g. image denoising with very deep networks [60]. Residuals were also used for pansharpening in [35], in the context of sparse representation, and in [61], where a residual-based regressor was proposed. 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 [59] 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 denosing, super-resolution, and pansharpening.

Very recently, two groups of researchers have proposed using residual learning for pansharpening [44], [42], [43], 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 [44] experiments are carried out on Landsat 7 images, with geometric and spectral characteristics very far from those of typical multiresolution images of interest. In [42], [43], 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

$$f(\mathbf{x}, \Phi) = f_L(f_{L-1}(\ldots f_1(\mathbf{x}, \Phi_1), \ldots, \Phi_{L-1}), \Phi_L) \quad (2)$$

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 to use 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 [62] 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 backpropagate easily through so many layers. A number of approaches have been proposed to deal with this problem, including 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 [63], 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 [64], thus removing

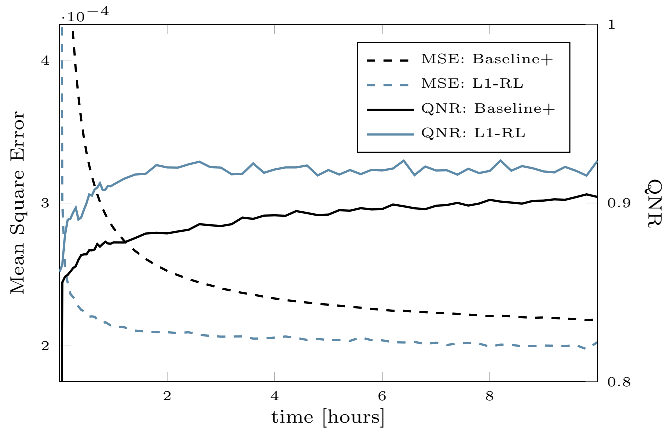


Fig. 6. Testing the coherence between reduced-resolution (MSE) and full-resolution (QNR) quality indicators.

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. 5, 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 at convergence over the 3-layer net. Finally, augmentation through radiometric indices, which was found beneficial 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, used on full-resolution data. As an example, for two different architectures, Fig. 6 shows the behavior of MSE and QNR (a full-resolution quality index) as training proceeds on the GeoEye-1 dataset. On a rough scale, results are consistent, with QNR approaching 1 (best quality) as the MSE reduces. However, even with stable MSE, significant fluctuations in QNR are observed, maybe due to the imperfect MTF modeling, which suggest due care when considering these measures.

In conclusion, these preliminary results allow us to proceed safely with design choices. Specifically, from now on we will focus on the residual learning architecture of Fig.4, with L1 loss and without input augmentation. Moreover, we will keep the original 3-layer CNN, as current experiments do not support the adoption of a deeper architecture[2].

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. We use the best architecture emerging from the analyses of previous Section, a three-layer CNN 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.

[2]This result, in partial disagreement with the current literature, may be due to a number of reasons, including insufficient training data. Therefore, we will keep considering this option in future work, and investigate it further.

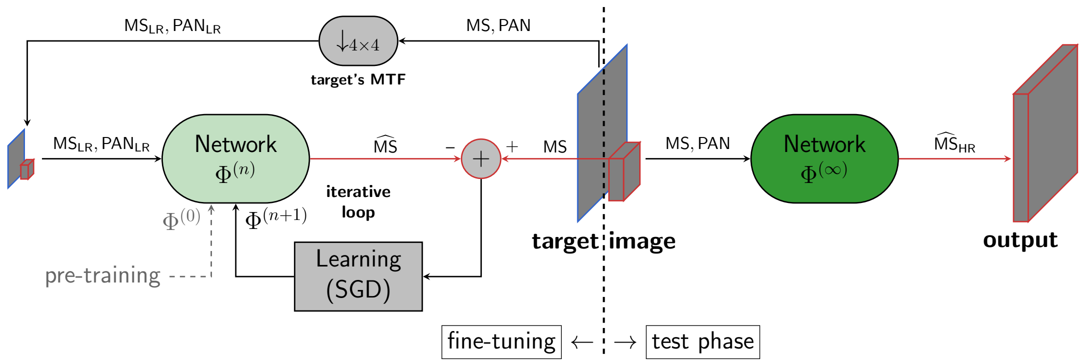


Fig. 7. Workflow of the proposed target-adaptive modality. The target image at center figure feeds the fine-tuning process on the left-hand side until convergence. Then, when fine-tuning is over, it enters the fully trained network on the right-hand side to produce the desired pansharpened output.

The whole process is described in more detail in Fig. 7. Initially, the network, represented by the light green oval[3] in the left part of the figure, uses the parameters, $\Phi^{(0)}$, selected in the pre-training phase. Then, these are iteratively fine-tuned, $\ldots, \Phi^{(n)}, \Phi^{(n+1)}, \ldots$, by training over data sampled from the target image itself, always using Wald's protocol (note that red lines convey only MS data). Upon convergence, the parameters are freezed to $\Phi^{(\infty)}$ and used in the final network, shown on the right as a dark green oval, to carry out actual pansharpening of the target image. Our experiments show that 50 iterations ensure a good quality fine-tuning in all cases, hence, to keep complexity under control, we use this default value rather than thresholding over the loss variations.

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 just 1.5 seconds on a GPU-equipped computer, and about 210 seconds on a general purpose CPU, which is appreciable by the user but certainly affordable in view of the ensuing performance improvement. Note also that the computational overhead for fine-tuning does not grow with the image size, as only a certain number of samples need be extracted from the target image to ensure a good adaptation. Hence, for large real-world images, it becomes negligible w.r.t. actual pansharpening even in the CPU-only case.

This very short processing time is readily explained, in fact

- the network is already pre-trained, and therefore fine-tuning requires a small number of iterations;
- the adaptation involves only the target data (possibly, only a subset of them) which are orders of magnitude less than the training data;
- the selected architecture, with residual learning, trains 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. 8 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-GE-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-WV-2) and target clips (Washington-WV-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-WV-2) and target clips (Adelaide-WV-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

[3]For simplicity *Network* includes also the MS upsampling (refer to Fig. 1).

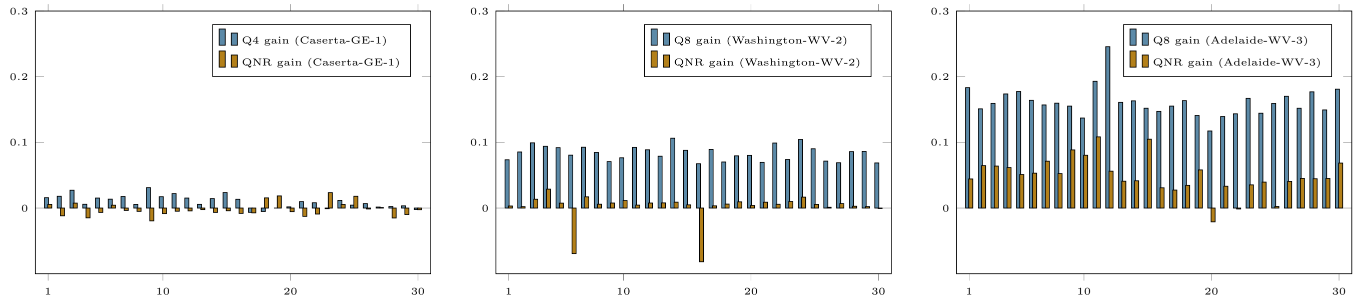


Fig. 8. Performance gain ensured by fine-tuning over 30 target clips. Q4/Q8: full-reference measure computed on reduced resolution data. QNR: no-reference measure computed on full resolution data. From left to right: favourable case (same sensor, same scene), typical case (same sensor, different scenes), challenging case (different sensors).

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 [65], [66], [67], 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;
- discuss the computational issue.

A. Methods under analysis

Our baseline is the PNN proposed in [41], with the input augmented by radiometric indexes. However, since all the solutions proposed here have been implemented in Python using Theano [68], we have re-implemented in this framework also the baseline, originally developed with Caffe [69], 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 [41]. 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], GSA [7], Indusion [13], AWLP [12], ATWT-M3 [10], MTF-GLP-HPM

TABLE I
CNN hyper-parameter for all proposed methods: # output features × # input channels × 2D filter support. Red entries only for baseline (input augmentation)

Sensor	ConvLayer 1	Activ.	ConvLayer 2	Activ.	ConvLayer 3
Ikonos	48×5/ 7×5×5	ReLU	32×48×5×5	ReLU	4×32×5×5
GeoEye-1	48×5/ 7×9×9	ReLU	32×48×5×5	ReLU	4×32×5×5
WorldView-2	48×9/13×9×9	ReLU	32×48×5×5	ReLU	8×32×5×5
WorldView-3	48×9/13×9×9	ReLU	32×48×5×5	ReLU	8×32×5×5

[15], MTF-GLP-CBD [16], BDSD [9], its recent extension C-BDSD [22], and SR [35], based on sparse representations. Details on these methods can be found in [28] and in [41], 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 upsampler. The software used to implement the methods and carry out all experiments is available online [70] to ensure full reproducibility.

B. Performance measures

To assess performance we use the framework made available online [47] by Vivone *et al.*, and described in [28]. 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 Wald's protocol [50], [10], [28], 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 [50] it is left as an open concern. Here, we adopt the solution proposed by Aiazzi *et al.* [16], and implemented in the Open Remote Sensing repository [47],

TABLE II

TEST SET AND TRAINING SET COMBINATIONS USED IN THE EXPERIMENTS.

Test dataset (# clips)	Training dataset	Op. condition	Table	Figure
Caserta-IK (50)	Caserta-IK	favourable	III	
Caserta-IK (50)	Caserta-GE-1	challenging	III	
Caserta-GE-1 (70)	Caserta-GE-1	favourable	IV	
Caserta-GE-1 (70)	Caserta-IK	challenging	IV	
Caserta-WV-2 (30)	Caserta-WV-2	favourable	V	
Stockholm-WV-2 (30)	Caserta-WV-2	typical	VI	9, 10
Washington-WV-2 (81)	Caserta-WV-2	typical	VII	9, 10
Adelaide-WV-3 (45)	Caserta-WV-2	challenging	VIII	11

[28], which uses a different low-pass filter for each MS band and for the PAN, matched to the specific channel MTF.

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

- **SAM**: Spectral Angle Mapper [53];
- **ERGAS**: Global adimens. relative synthesis error [71];
- **Q**: Average universal image Quality index [72];
- **Q4 / Q8**: 4 / 8-band extension of **Q** [73];

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 [74];
- $\mathbf{D}_\lambda$: Spectral component of QNR;
- $\mathbf{D}_S$: 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, which performs only MS upsampling, the $\mathbf{D}_\lambda$ 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, $\mathbf{D}_S$ 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 (IK for short), GeoEye-1 (GE-1), WorldView-2 (WV-2) and WorldView-3 (WV-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 (IK) to 0.31m (WV-3), while 4 (IK, GE-1) to 8 (WV) bands are available, covering the visible and near-infrared regions of the spectrum.

TABLE III

PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE CASERTA-IK DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q4	**Q**	**SAM**	**ERGAS**	$\mathbf{D}_\lambda$	$\mathbf{D}_S$	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.448	.705	3.009	2.884	0	.0438	.956
PRACS	.659	.802	2.994	2.360	.0493	.1148	.842
GSA	.658	.777	3.468	2.433	.1061	.1883	.729
Indusion	.593	.766	3.280	2.796	.1264	.1619	.734
AWLP	.714	.839	**2.843**	2.113	.1384	.1955	.695
ATWT-M3	.558	.725	3.581	3.033	.1244	.1452	.749
MTF-GLP-HPM	.718	.842	2.882	2.055	.1524	.2186	.665
MTF-GLP-CBD	.713	.838	2.909	2.210	.0750	.0957	.837
BDSD	.720	**.858**	2.914	**1.985**	**.0395**	.0884	.876
C-BDSD	**.720**	.857	2.910	2.055	.0710	.1218	.817
SR	.667	.836	3.109	2.148	.0691	**.0264**	**.906**
Baseline (Theano)	.769	.906	2.200	1.588	.0542	.0698	.880
L1	.781	.912	2.074	1.507	.0285	.0676	.906
L1-RL	.787	.916	2.022	1.470	.0220	**.0635**	.916
L1-RL-FT	**.797**	**.923**	**1.939**	**1.401**	**.0207**	.0641	**.917**
L1-RL (cross)	.674	.832	3.016	2.334	.0249	.0708	.878
L1-RL (cross)-FT	.775	.908	2.162	1.532	.0249	.0687	.908

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 [41]. More typpcally, training and test data are expected to be unrelated. For example, a network trained on the Caserta-WV-2 scene may be used for the pansharpening of the WV-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-WV-2 scene, may be used to pansharpen a WV-3 image. In Tab. II, 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 33×33 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 [41] where additional details can be found. Eventually, the trained nets are tested on a number of 1280×1280 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 IK, GE-1, and WV-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.

TABLE IV
PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE CASERTA-GE-1 DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q4	**Q**	**SAM**	**ERGAS**	**D_λ**	**D_S**	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.518	.818	2.339	1.816	0	.0832	.917
PRACS	.699	.857	3.236	2.429	**.0470**	.0877	.870
GSA	.687	.814	4.336	2.855	.1120	.1826	.727
Indusion	.574	.777	3.536	3.548	.1270	.1262	.765
AWLP	.717	.861	3.629	2.613	.1257	.1521	.743
ATWT-M3	.601	.791	3.554	3.073	.0712	.0710	.863
MTF-GLP-HPM	.736	.872	**3.220**	5.034	.1526	.1815	.695
MTF-GLP-CBD	.716	.848	3.559	2.863	.0706	.0838	.851
BDSD	**.740**	**.883**	3.338	**2.234**	.0490	.0994	.857
C-BDSD	.739	.878	3.482	2.437	.0832	.1342	.795
SR	.686	.864	3.672	2.527	.0508	**.0288**	**.922**
Baseline (Theano)	.815	.943	2.101	1.496	.0387	.0615	.903
L1	.822	.945	2.069	1.459	**.0250**	.0584	.918
L1-RL	.820	.946	2.058	1.480	.0296	**.0485**	**.923**
L1-RL-FT	**.829**	**.951**	**2.007**	**1.414**	.0273	.0523	.922
L1-RL (cross)	.654	.837	3.449	2.659	.0478	.1077	.850
L1-RL (cross)-FT	.805	.940	2.274	1.582	.0286	.0585	.915

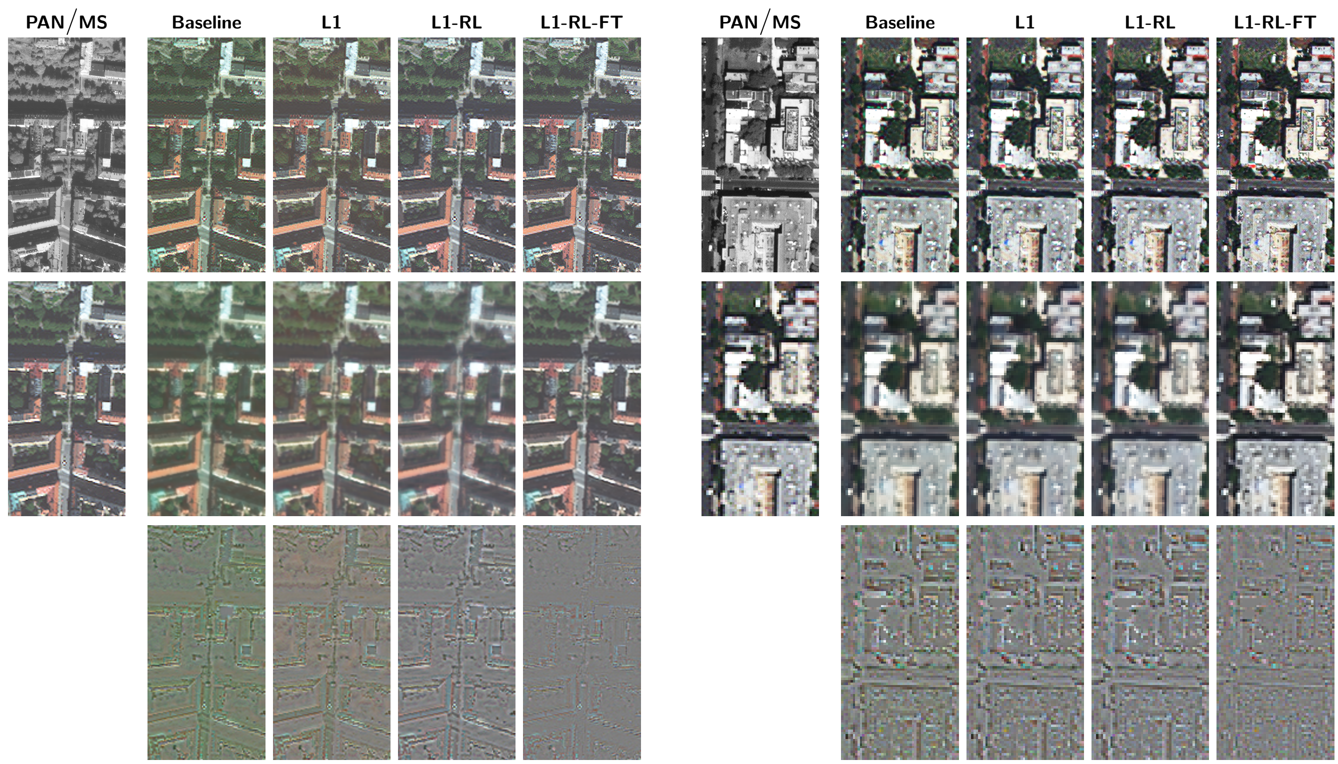
TABLE V
PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE CASERTA-WV-2 DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q8	**Q**	**SAM**	**ERGAS**	**D_λ**	**D_S**	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.496	.712	4.944	5.776	0	.0653	.935
PRACS	.791	.879	3.699	2.410	**.0234**	.0734	.905
GSA	.771	.849	4.530	2.761	.0427	.0836	.877
Indusion	.693	.837	3.726	3.202	.0552	.0649	.884
AWLP	.813	.904	**3.418**	2.256	.0665	.0849	.855
ATWT-M3	.704	.817	4.065	3.161	.0675	.0748	.863
MTF-GLP-HPM	**.824**	**.908**	3.450	**2.092**	.0755	.0953	.837
MTF-GLP-CBD	.812	.898	3.583	2.357	.0463	.0605	.896
BDSD	.811	.905	3.745	2.264	.0483	.0382	.916
C-BDSD	.800	.895	3.989	2.636	.0251	.0458	.930
SR	.754	.882	4.456	2.682	.0284	**.0325**	**.940**
Baseline (Theano)	.868	.954	2.342	1.434	.0195	.0464	.935
L1	.871	.956	2.306	1.422	**.0164**	.0489	.935
L1-RL	.872	.958	2.293	1.438	.0238	**.0404**	**.937**
L1-RL-FT	**.877**	**.961**	**2.191**	**1.368**	.0216	.0428	.937

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 [41], that PNN improves significantly and almost uniformly over all measures with respect to reference methods. Even the sparse-representation method, SR, which learns a dedicated dictionary for each target image, is only competitive on no-reference indicators, while showing large performance losses on full-reference ones. 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 $\mathbf{D}_\lambda$, 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 VI and VII results for WV-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 PRACS, C-BDSD and SR on no-reference measures, and MTF-GLP-CBD on full-reference measures. On Stockholm, in particular, a large $\mathbf{D}_\lambda$ 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. 9 (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 data, pansharpened images are affected by a large spectral error, both al 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 also this problem, providing satisfactory results in terms of both



(a) (b)

Fig. 9. Output of baseline and proposed methods on clips from the Stockholm-WV-2 (a) and Washington-WV-2 (b) images. From top to bottom: PAN + results at full resolution, MS + results at reduced resolution, error maps for the reduced-resolution case.

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. 10 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-CBD, Indusion, AWLP. PRACS and ATWT-M3, instead, exhibit oversmoothing, while C-BDSD and ATWT-M3 are affected by spectral distortion. Finally, SR provides sharp results but with some spatial artifacts which are clearly visible on the shadowed areas. At reduced resolution, the proposed method seems 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

TABLE VI
PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE STOCKHOLM-WV-2 DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q8	**Q**	**SAM**	**ERGAS**	$\mathbf{D}_\lambda$	$\mathbf{D}_S$	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.369	.592	8.401	9.935	0	.0522	.948
PRACS	.696	.791	7.997	7.041	**.0126**	.0935	**.895**
GSA	.710	.795	9.013	7.097	.0425	.1341	.830
Indusion	.628	.767	8.721	7.650	.0589	.1084	.840
AWLP	.749	.839	7.353	6.031	.0624	.1166	.829
ATWT-M3	.634	.730	8.827	8.237	.0849	.1311	.796
MTF-GLP-HPM	.758	.849	7.103	5.699	.0749	.1316	.805
MTF-GLP-CBD	**.771**	**.862**	**6.746**	**5.657**	.0427	.0671	.893
BDSD	.743	.846	8.305	6.055	.0820	.1544	.776
C-BDSD	.743	.848	8.130	6.545	.0543	.0634	.886
SR	.726	.848	8.545	6.073	.0647	**.0466**	.894
Baseline (Theano)	.731	.854	7.4553	5.582	.1064	.0444	.854
L1	.649	.823	8.004	6.052	.0724	.0503	.881
L1-RL	.667	.810	6.995	6.300	.0496	**.0438**	**.909**
L1-RL-FT	**.833**	**.914**	**4.826**	**3.724**	**.0427**	.0713	.890

and without fine-tuning. In detail, for Tab. III, concerning the Caserta-IK image, the network is trained on Caserta-GE-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

Fig. 10. Output of proposed and reference methods on clips from the Stockholm-WV-2 (top) and Washington-WV-2 (bottom) images. PAN + results at full resolution (1st and 4th rows), MS + results at reduced resolution (2nd and 5th rows), error maps for the reduced-resolution case (3rd and 6th rows).

TABLE VII
PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE WASHINGTON-WV-2 DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q8	**Q**	**SAM**	**ERGAS**	**D_λ**	**D_S**	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.383	.615	7.496	7.179	0	.0578	.942
PRACS	.667	.774	7.404	5.359	**.0124**	.0950	.894
GSA	.703	.766	8.970	5.380	.0445	.1444	.818
Indusion	.662	.783	7.530	5.585	.0704	.1090	.829
AWLP	.753	.842	**7.089**	4.570	.0744	.1276	.808
ATWT-M3	.602	.716	8.203	6.124	.0666	.1348	.808
MTF-GLP-HPM	**.767**	**.849**	7.127	**4.451**	.0831	.1422	.787
MTF-GLP-CBD	.763	.846	7.523	4.687	.0568	.0814	.866
BDSD	.750	.844	8.087	4.644	.0782	.1100	.821
C-BDSD	.755	.844	8.286	4.951	.0471	.0424	.913
SR	.723	.833	8.698	4.855	.0463	**.0423**	**.914**
Baseline (Theano)	.789	.885	5.431	3.561	.0441	.0634	.895
L1	.780	.882	5.645	3.593	.0426	.0636	.897
L1-RL	.780	.884	5.520	3.660	.0441	.0507	.907
L1-RL-FT	**.857**	**.929**	**4.654**	**2.793**	**.0416**	**.0498**	**.911**

TABLE VIII
PERFORMANCE INDICATORS AT REDUCED (FULL-REFERENCE) AND FULL (NO-REFERENCE) RESOLUTION ON THE ADELAIDE-WV-3 DATASET.

	FULL-REFERENCE				NO-REFERENCE		
	Q8	**Q**	**SAM**	**ERGAS**	**D_λ**	**D_S**	**QNR**
(desired value)	(1)	(1)	(0)	(0)	(0)	(0)	(1)
EXP	.397	.651	7.410	7.773	0	.0716	.928
PRACS	.716	.831	7.351	5.271	**.0169**	.1018	.883
GSA	.731	.830	8.407	5.294	.0541	.1457	.808
Indusion	.658	.803	7.585	6.024	.0564	.1170	.833
AWLP	.763	.863	7.304	4.874	.0578	.1069	.842
ATWT-M3	.621	.761	7.992	6.383	.0765	.1052	.826
MTF-GLP-HPM	.778	.876	**7.062**	5.155	.0795	.1291	.802
MTF-GLP-CBD	.788	.880	7.187	4.610	.0545	.0620	.887
BDSD	.788	**.883**	7.422	**4.501**	.0441	.0764	.883
C-BDSD	**.788**	.882	7.755	5.487	.0297	.0532	.919
SR	.757	.874	7.929	4.683	.0463	**.0237**	**.932**
Baseline (Theano)	.743	.864	8.182	4.963	.0702	.0641	.870
L1 (cross)	.606	.837	9.544	5.915	.0740	.0649	.866
L1-RL (cross)	.694	.836	7.787	5.390	.0894	**.0473**	.867
L1-RL (cross)-FT	**.855**	**.938**	**5.093**	**3.256**	**.0393**	.0474	**.915**

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 WV-3 sensor, using the network trained on WV-2 data. In this case there is no same-sensor comparison, we just used the available WV-2 net. Again, after fine-tuning, the proposed method works much better than conventional methods for full-reference measures, and only slightly worse than C-BDSD and SR for full-resolution measures. The visual inspection of Fig. 11, however, shows that SR exhibits annoying spatial patterns at full-resolution, while C-BDSD is affected by spectral distortion. The large performance gain w.r.t. all other methods appears also obvious from the analysis of error images, since all reference methods present strong image-related structures (spatial distortion) and colored regions (spectral distortion). Overall, thanks to fine-tuning, the proposed method seems preferable to all references even in a cross-sensor setting.

Our analysis completes with some notes on computational complexity. On a GPU-equipped computer (GeForce GTX Titan X, Maxwell, 12GB) the proposed method runs very fast, taking little more than 1 second/Mpixel on the average, including fine-tuning.

Needless to say, this is the natural configuration for any method based on deep learning. On the other hand, the algorithm can also run on a simple CPU-equipped computer, in which case complexity may become an issue. Therefore, for all methods, we measured average CPU times (Intel Xeon E5-2670 1.80GHz, 64GB), reporting results in Tab. IX. For 1280×1280-pixel clips, running times go (excluding SR) from the 1.4 s/clip of Indusion and BDSD to the 17.3 s/clip of ATWT-M3. Without fine-tuning, PNN-L1-RL is only somewhat slower, 24.4 s/clip, which is quite reasonable given the huge number of convolutions carried out in a CNN. Of course, the iterative fine-tuning process, FT in the table, is much heavier, adding 210 seconds to the total CPU-time. Whether this may prevent use of the proposed method depends only on user requirements. On the other hand, SR suffers from the very same problem, due to on-line dictionary learning, and is even slower.

TABLE IX
CPU TIMES (S) FOR 1280×1280 AND 2560×2560-PIXEL CLIPS.

PAN size	PRACS	GSA	Indusion	AWLP	ATWT-M3	MTF-*	BDSD	C-BDSD	SR	PNN-L1-RL	FT
1280^2	10.7	1.7	1.4	6.3	17.3	2.5	1.4	5.9	316.4	24.4	210.1
2560^2	43.2	6.9	5.2	23.2	69.7	10.0	5.4	23.7	3381.7	102.3	209.6

Turning to 2560×2560-pixel clips, we notice that for almost all methods[4], including PNN, CPU times scale about linearly with size. However, the cost of fine-tuning does not increase, since a good adaptation to the target image can be achieved based on a suitable subset of the image. Therefore, its weight on the overall cost decreases, becoming eventually negligible for the large-size images, say, 20000×20000 pixels, used in real-world practice.

VI. CONCLUSIONS

We started from our recently proposed [41] 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.

[4]We selected SR parameters to optimize performance, whatever the CPU-time. Other choices are possible, but this goes beyond the scope of this work.

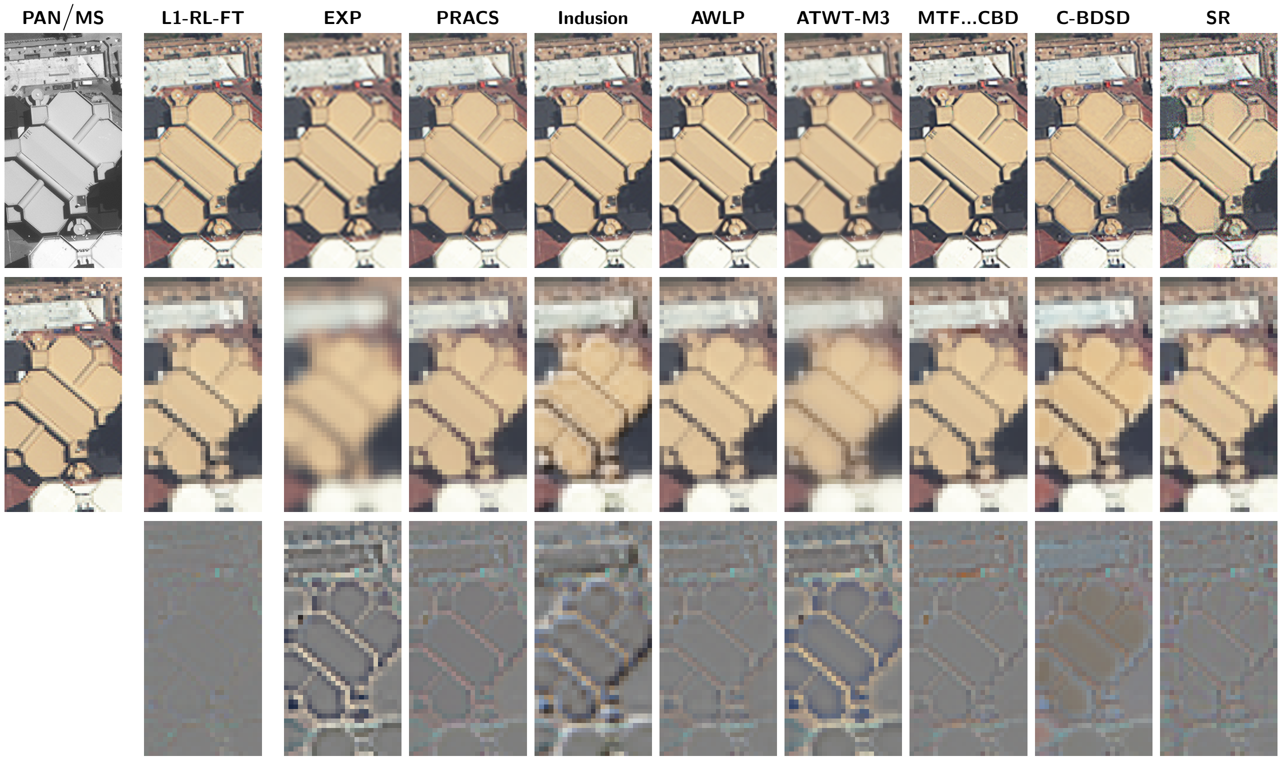


Fig. 11. Output of proposed and reference methods on clips from the Adelaide-WV-3 image. From top to bottom: PAN + results at full resolution, MS + results at reduced resolution, error maps for the reduced-resolution case.

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, with a reasonable computational cost and 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 [75], 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.

VII. Acknowledgement

We thank the Authors of all reference methods for sharing their codes, and those of the Open RS platform for providing the code for accuracy evaluation. We also thank DigitalGlobe for providing the images used for training and testing our methods.

References

[1] V. Shettigara, “A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set,” *Photogramm. Eng. Remote Sens.*, 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.

[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 Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 309–312, 2004.

[4] P. Chavez and A. Kwarteng, “Extracting spectral contrast in Landsat thematic mapper image data using selective principal component analysis,” *Photogramm. Eng. Remote Sens.*, vol. 55, no. 3, pp. 339–348, 1989.

[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. 343–365, 1987.

[6] C. Laben. and B. Brower, “Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening,” *U.S. Patent 6011875, 2000.*, 2000.

[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 pan sharpening 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,” *Photogramm. Eng. Remote Sens.*, 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 Geosci. Remote Sens. Lett.*, 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," *Photogramm. Eng. Remote Sens.*, vol. 72, no. 5, pp. 591–596, 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 Trans. Geosci. Remote Sens.*, 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 Trans. Geosci. Remote Sens.*, 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] L. Zhang, H. Shen, W. Gong, and H. Zhang, "Adjustable model-based fusion method for multispectral and panchromatic images," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 42, no. 6, pp. 1693–1704, Dec 2012.

[22] A. Garzelli, "Pansharpening of multispectral images based on nonlocal parameter optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2096–2107, April 2015.

[23] X. Meng, H. Shen, H. Li, Q. Yuan, H. Zhang, and L. Zhang, "Improving the spatial resolution of hyperspectral image using panchromatic and multispectral images: An integrated method," in *WHISPERS*, June 2015.

[24] H. Shen, X. Meng, and L. Zhang, "An integrated framework for the spatio-temporal-spectral fusion of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7135–7148, Dec 2016.

[25] S. Zhong, Y. Zhang, Y. Chen, and D. Wu, "Combining component substitution and multiresolution analysis: A novel generalized BDSD pansharpening algorithm," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 10, no. 6, pp. 2867–2875, June 2017.

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

[27] 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 Sens.*, vol. 125, pp. 78–105, 2017.

[28] 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.

[29] 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 Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 308–319, Jan 2017.

[30] 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.

[31] 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.

[32] 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.

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

[34] M. Cheng, C. Wang, and J. Li, "Sparse representation based pansharpening using trained dictionary," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 293–297, 2014.

[35] M. R. Vicinanza, R. Restaino, G. Vivone, M. D. Mura, 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.

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

[37] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv:1508.00092*, 2015.

[38] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel, "Land cover classification via multitemporal spatial data by deep recurrent neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1685–1689, Oct 2017.

[39] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: a review," *arXiv:1710.03959*, 2017.

[40] 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.

[41] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sensing*, vol. 8, no. 7, 2016.

[42] Y. Wei and Q. Yuan, "Deep residual learning for remote sensed imagery pansharpening," in *RSIP*, May 2017.

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

[44] Y. Rao, L. He, and J. Zhu, "A residual convolutional neural network for pan-sharpening," in *RSIP*, May 2017.

[45] A. Azarang and H. Ghassemian, "A new pansharpening method using multi resolution analysis framework and deep neural networks," in *2017 3rd Int. Conf. on Pattern Recognition and Image Analysis (IPRIA)*, April 2017.

[46] C. Dong, 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.

[47] "Open Remote Sensing," http://openremotesensing.net, (accessed on 5 May 2017).

[48] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "CNN-based pansharpening of multi-resolution remote-sensing images," in *Joint Urban Remote Sensing Event (JURSE)*, Dubai, March 2017.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1106–1114.

[50] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolution: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, pp. 691–699, 1997.

[51] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning." in *Int. Conf. on Machine Learning (ICML)*, Feb 2013, pp. 1139–1147.

[52] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

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

[54] C.-I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1927–1932, Aug 2000.

[55] R. Girshick, "Fast R-CNN," in *2015 IEEE Int. Conf. on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[56] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*. Springer, 2012, pp. 711–730.

[57] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *2013 IEEE Int. Conf. on Computer Vision (ICCV)*, Dec 2013, pp. 1920–1927.

[58] J. K. L. J. Kim and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654.

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

[60] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[61] Q. Wang, W. Shi, and P. M. Atkinson, "Area-to-point regression kriging for pan-sharpening," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 151–165, 2016.

[62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[64] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.
[65] M. M. Khan, L. Alparone, and J. Chanussot, "Pansharpening quality assessment using the modulation transfer functions of instruments," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3880–3891, Nov 2009.
[66] B. Aiazzi, L. Alparone, S. Baronti, R. Carlà, A. Garzelli, and L. Santurri, "Full scale assessment of pansharpening methods and data products," in *Proc. SPIE*, vol. 9244, 2014.
[67] F. Palsson, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Quantitative quality evaluation of pansharpened imagery: Consistency versus synthesis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1247–1259, March 2016.
[68] "Theano documentation webpage," http://deeplearning.net/software/theano, (accessed on 5 May 2017).
[69] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014.
[70] "GRIP home page," http://grip.unina.it, (accessed on 5 May 2017).
[71] L. Wald, "Data fusion: Definitions and architectures–fusion of images of different spatial resolutions," *Les Presses de l'Ècole des Mines*, 2002.
[72] Z. Wang and A. Bovik, "A universal image quality index," *IEEE Signal Proc. Lett.*, vol. 9, no. 3, pp. 81–84, March 2002.
[73] L. Alparone, S. Baronti, A. Garzelli, and F. Nencini, "A global quality measurement of pan-sharpened multispectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 313–317, Oct 2004.
[74] 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.
[75] S. Lolli, L. Alparone, A. Garzelli, and G. Vivone, "Haze correction for contrast-based multispectral pansharpening," *IEEE Geosci. Remote Sens. Lett.*, in press, 2017.