remote sensing

MDPI

Article

Fast Full-Resolution Target-Adaptive CNN-Based Pansharpening Framework

Matteo Ciotola [1,*] and Giuseppe Scarpa [2]

[1] Department of Electrical Engineering and Information Technology (DIETI), University Federico II, 80125 Naples, Italy
[2] Department of Engineering, University Parthenope, Centro Direzionale ISOLA C4, 80133 Naples, Italy
* Correspondence: matteo.ciotola@unina.it

Abstract: In the last few years, there has been a renewed interest in data fusion techniques, and, in particular, in pansharpening due to a paradigm shift from model-based to data-driven approaches, supported by the recent advances in deep learning. Although a plethora of convolutional neural networks (CNN) for pansharpening have been devised, some fundamental issues still wait for answers. Among these, cross-scale and cross-datasets generalization capabilities are probably the most urgent ones since most of the current networks are trained at a different scale (reduced-resolution), and, in general, they are well-fitted on some datasets but fail on others. A recent attempt to address both these issues leverages on a target-adaptive inference scheme operating with a suitable full-resolution loss. On the downside, such an approach pays an additional computational overhead due to the adaptation phase. In this work, we propose a variant of this method with an effective target-adaptation scheme that allows for the reduction in inference time by a factor of ten, on average, without accuracy loss. A wide set of experiments carried out on three different datasets, GeoEye-1, WorldView-2 and WorldView-3, prove the computational gain obtained while keeping top accuracy scores compared to state-of-the-art methods, both model-based and deep-learning ones. The generality of the proposed solution has also been validated, applying the new adaptation framework to different CNN models.

Keywords: data fusion; multiresolution analysis; super-resolution; pansharpening

check for updates

Citation: Ciotola, M.; Scarpa, G. Fast Full-Resolution Target-Adaptive CNN-Based Pansharpening Framework. *Remote Sens.* **2023**, *15*, 319. https://doi.org/10.3390/rs15020319

Academic Editor: Amin Beiranvand Pour

Received: 24 November 2022
Revised: 23 December 2022
Accepted: 3 January 2023
Published: 5 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to the increasing number of remote sensing satellites and to renewed data sharing policies, e.g., the European Space Agency (ESA) Copernicus program, the remote sensing community calls for new data fusion techniques for such diverse applications as cross-sensor [1–4], cross-resolution [5–8] or cross-temporal [4,9,10] ones, for analysis, information extraction or synthesis tasks. In this work, we target the pansharpening of remotely sensed images, which amounts to the fusion of a single high resolution panchromatic (PAN) band with a set of low resolution multispectral (MS) bands to provide a high-resolution MS image.

A recent survey [11] gathered the available solutions in four categories: component substitution (CS) [12–15], multiresolution analysis (MRA) [16–19], variational optimization (VO) [20–23], and machine/deep learning (ML) [24–28]. In the CS approach, the multispectral image is transformed in a suitable domain, one of its components is replaced by the spatially rich PAN and the image is transformed back into the original domain. For example, in the simple case where only three spectral bands are concerned, the Intensity-Hue-Saturation (IHS) transform can be used for this purpose. The same method has been straightforwardly generalized to the case of a larger number of bands in [13]. Other examples of this approach, to mention a few, are whitening [12], Brovey [14] or the Gram–Schmidt decomposition [15]. In the MRA approach, instead, pansharpening is addressed resorting to a multi-resolution decomposition, such as decimated or undecimated wavelet transforms [16,18] and Laplacian pyramids [17,19], for proper extraction of the detail component to be injected into the resized

multispectral component. VO approaches leverage on suitable acquisition or representation models to define a target function to optimize. This can involve the degradation filters mapping high-resolution to low-resolution images [22], sparse representation of the injected details [29], probabilistic models [21] and low-rank PAN-MS representations [23]. Needless to say, the paradigm shift from model-based to ML approaches registered in the last decade has also heavily had an impact on such diverse remote sensing-related image processing problems as classification, detection, denoising, data fusion and so forth. In particular, the first pansharpening convolutional neural network (PNN) was introduced by Masi et al. (2016) [24], and it was rapidly followed by many other works [25,27,28,30–32].

It seems safe to say that deep learning is currently the most popular approach for pansharpening. Nonetheless, it suffers from a major problem: the lack of ground truth data for supervised training. Indeed, multiresolution sensors can only provide the original MS-PAN data, downgraded in space or spectrum, never their high-resolution versions, which remain to be estimated. The solution to this problem introduced in [24], and still adopted by many others, consists in a resolution shift. The resolution of the PAN-MS data is properly downgraded by a factor equal to the PAN-MS resolution ratio in order to obtain input data whose ground-truth (GT) is given by the original MS. Any network can therefore be trained in a fully supervised manner, although in a lower-resolution domain, and then be used on full-resolution images at inference time. The resolution downgrade paradigm is not new, as it stems from Wald's protocol [33], a procedure employed in the contest of the pansharpening quality assessment, and presents two main drawbacks:

i. It requires the knowledge of the point spread function (also referred to as sensor Modulation Transfer Function, MTF, in the pansharpening context), which characterizes the imaging system, to apply before decimation to obtain the reduced resolution dataset;
ii. It relies on a sort of scale-invariance assumption (a method optimized at reduced resolution is expected to work equally well at full resolution).

In particular, the latter limitation has recently motivated several studies aimed at circumventing the resolution downgrade [31,32,34,35]. These approaches resort to losses that do not require any GT being oriented to consistency rather than to the synthesis assessment. During training, the full-resolution samples feed the network whose output is then compared to the two input components, MS and PAN, once suitably reprojected in their respective domains. The way such reprojections are realized, in combination with the measurement employed, i.e., the *consistency* check, has been the object of intense research since the seminal Wald's paper [33], and is still an open problem. In addition, a critical issue is also represented by the lack of publicly available datasets that are sufficiently large and representative to ensure generality to the trained networks. A solution to this, based on the target-adaptivity principle, was proposed in [36] and later adopted in [34,35] too. On the downside, target-adaptive models pay a computational overhead at inference time, which increases when operating at full-resolution, as occurs in [35].

Motivated by the above considerations, following the research line drawn in [35], which combines full-resolution training and target-adaptivity, in this work, we introduce a new target-adaptive scheme, which allows reducing the computational overhead while preserving the accuracy of the pansharpened products. Experiments carried out on GeoEye-1, WorldView-2 and WorldView-3 images demonstrate the effectiveness of the proposed solution, achieving computational gains of about one order of magnitude ($\sim$10 times faster on average) for fixed accuracy levels.

In the next Section 2, we describe the related work and the proposed solution. Then, we provide experimental results in Section 3 and a related discussion in Section 4, before providing conclusions in Section 5.

2. Materials and Methods

Inspired by the recent paper [35], we propose here a new method that reduces the computational burden at inference time when using target-adaptivity. The adaptation overhead, in fact, may be not negligible even if relatively shallow networks such as [24,25,36–38] are used.

Likewise [35], here, we also propose a framework rather than a single model. In particular, we use the "zoom" (Z) versions [35] of A-PNN [36], PanNet [25] and DRPNN [30] as baselines, which will be referred to as Z-PNN, Z-PanNet and Z-DRPNN, respectively. However, for the sake of simplicity, the proposed solution will be presented with respect to a fixed model, Z-PNN, without lack of generality.

2.1. Revisiting Z-PNN

The method proposed in [35], called Zoom Pansharpening Neural Network (Z-PNN), inherits the target-adaptive paradigm introduced in [36], but it is based on a new loss conceived to use full-resolution images without ground-truths, hence allowing a fully unsupervised training and tuning. In Figure 1, the top-level flowchart of the full-resolution target-adaptive inference scheme is shown. The target-adaptive modality involves two stages: adaptation and prediction. The former is an iterative tuning phase where the whole PAN-MS input pair feeds the pre-trained network with initial parameters $\phi^{(0)}$. At each t-th iteration, the spatial and spectral consistencies of the output are quantified to provide updated parameters ($\phi^{(t)}$), thanks to a suitable composite loss that makes use of the input PAN and MS as references, respectively. After a predefined (by the User) number of iterations, the network parameters are frozen ($\phi^{(\infty)}$) and used for the last run of the network to provide the final pansharpened image. The default number of iterations is fixed to 100.

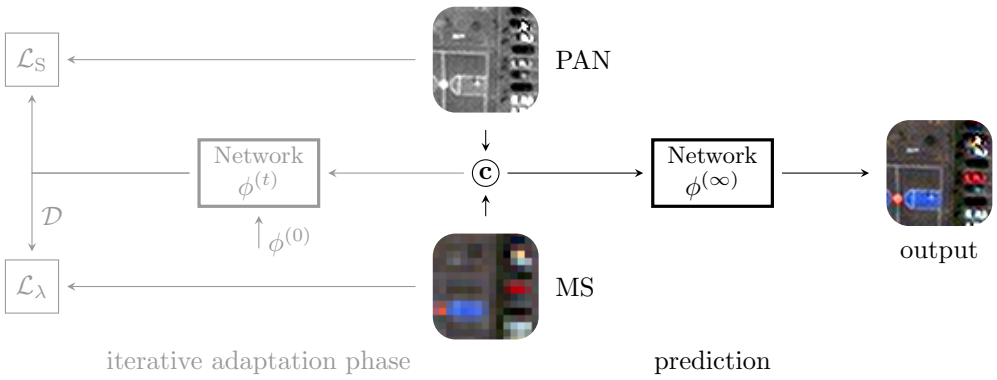


Figure 1. Target-adaptive inference scheme Z-<Network Name>. Available for networks (A-)PNN, PanNet and DRPNN.

More in detail, said M and P are the MS and PAN input components, respectively, $\widehat{M}$ the pansharpened image and $\mathcal{D}$ a suitable scaling operator. The overall loss is given by

$$\mathcal{L}(M, P; \widehat{M}) = \mathcal{L}_\lambda(M; \mathcal{D}(\widehat{M})) + \beta \mathcal{L}_S(P; \widehat{M}), \tag{1}$$

where the two loss terms are weighted by β to account for both spectral and spatial consistency. The spectral loss $\mathcal{L}_\lambda$ is given by

$$\mathcal{L}_\lambda = \| \mathcal{D}(\widehat{M}) - M \|_1 \tag{2}$$

where $\| \cdot \|_1$ indicates the ℓ_1-norm, and the low-resolution projection operator $\mathcal{D}$ consists of band-dependent low-pass filtering, followed by spatial decimation at pace $R = 4$:

$$\mathcal{D}(M(\cdot, \cdot, b)) = [M(\cdot, \cdot, b) * h_b] \downarrow_R, \tag{3}$$

being h_b a suitable band-dependent point spread function.

On the other hand, the spatial loss term $\mathcal{L}_S$ is given by

$$\mathcal{L}_S(P; \widehat{M}) = 1 - \mathrm{E}_{i,j,b}\left[\min(\rho^{\max}_{i,j,b}, \rho^{\sigma}_{i,j,b}(\widehat{M}; P))\right] \tag{4}$$

where $\rho^{\sigma}_{i,j,b}(\widehat{M};P)$ is the correlation coefficient computed in a $\sigma\times\sigma$ window centered at location (i,j), between the b-th band of $\widehat{M}$ and P, and $\rho^{\max}_{i,j,b}$ is an upper bound correlation field estimated on a smoothed version of P and the 23-tap polynomial interpolation of M. By doing so, image locations where the correlation coefficient between $\widehat{M}$ and P have reached the corresponding bounding level and do not contribute to the gradient computation. Experimental evidence suggested setting $\sigma = R = 4$.

2.2. Proposed Fast CNN-Based Target-Adaptation Framework for Pansharpening

The core CNN architecture of Z-PNN is the model A-PNN summarized in Figure 2. It is relatively shallow in order to keep the computational overhead limited due to the adaptation phase (Figure 1). Nonetheless, such an overhead remains a serious issue that may prevent the use of this approach in critical situations when large-scale target images are concerned. As an example, on a commercial GPU, such as NVIDIA Quadro P6000, a 2048 $\times$ 2048 image (at PAN resolution) requires about a second per tuning iteration that sums up to a couple of minutes if hundred adaptation iterations are concerned. Such a timing scales about linearly with the image size as long as the GPU memory is capable at hosting the whole process, but once such capacity is exceeded, the computational burden grows much more quickly. Motivated by this consideration, we have explored a modified adaptation protocol still operating at full resolution and valid for generic pansharpening networks. The basic conjecture from which we have started is that the required adaptation is mainly due to the different atmospheric and geometrical conditions that can determine a mismatch between the training and the test datasets, whereas the intra-image spatial variability counts much less.

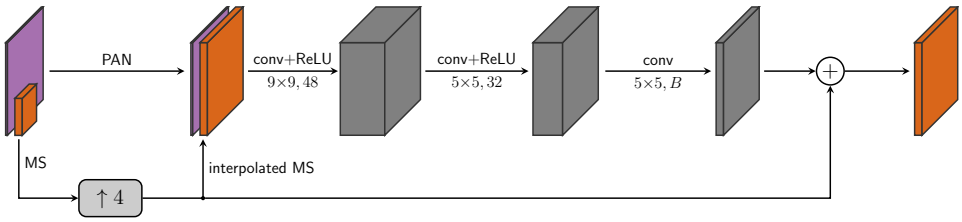
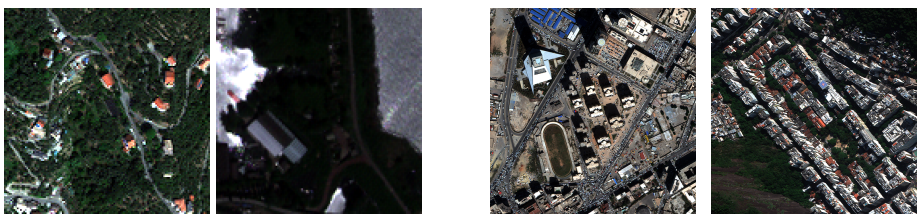


Figure 2. A-PNN model for Z-PNN.

Atmospheric and/or geometrical mismatches between training and test datasets can occur easily. Fog, pollution and daylight conditions are typical examples of atmospheric parameters that can heavily impact image features such as contrast and intensity. Figure 3a shows two details of GeoEye-1 images with similar content (forest) but taken from different acquisitions. The two details show different contrast and brightness levels, mostly because of a different light condition. On the other hand, geometry (think of satellite viewing angle) also plays an important role. In fact, scenes acquired at Nadir present geometries that differ from those visible off-Nadir. Urban settlement orientations with respect to the satellite orbit can also have some relevance. Figure 3b shows another pair of crops, showing some buildings lying on WorldView-3 images acquired with different viewing angles. The buildings clearly show different skews. It goes without saying that data augmentation techniques can only partially address these kinds of data issues.



(**a**) (**b**)

Figure 3. Atmospheric (**a**) and geometrical (**b**) mismatch examples.

On the basis of the above considerations, it makes sense to run the tuning iterations on partial tiles of the target image. In particular, we start from a small (128 × 128) central crop of the image and run half of the prefixed total number of iterations (fixed to 256). Once this first burst of 128 iterations is completed, the crop size is doubled in each direction (keeping the central positioning), while a halved (64) number of iterations are queued to the tuning process. The process goes on with this rule until the current crop covers the whole image. Depending on the image size, a residual number of iterations are needed to reach the total number of 256 iterations. This residue, in case, can be completed on the crop corresponding to the whole image. This rule is summarized in Table 1.

Table 1. Proposed target-adaptive iterations distribution. Bold numbers indicate the last phase (early stop) for the lighter version.

Shorter Image Side	Crop Size							
	128^2	256^2	512^2	1024^2	2048^2	4096^2	···	Whole
⋮								⋮
512–1023	128	**64**	32	-	-	-		32
1024–2047	128	64	**32**	16	-	-		16
2048–4095	128	64	32	**16**	8	-		8
4096–8191	128	64	32	16	**8**	4		4
⋮								⋮
Phase	1	2	3	4	5	6		(residue)

Besides this proposed method, hereinafter referred to as *Fast* Z-<Network Name>, we also test a lighter version, namely, *Faster* Z-<Network Name>, that undergoes an early stop of the tuning that allows skipping of the heaviest iterations that will involve the whole image (bold numbers in Table 1 correspond to the last tuning phase for the *Faster* version). In particular, for the sample network of Figure 2, the two versions will be called Fast Z-PNN and Faster Z-PNN, respectively. In addition to this, the "Z" versions of PanNet [25] and DRPNN [30], i.e., with full-resolution target-adaptation and loss (1), will also be tested.

3. Results

To prove the effectiveness of the proposed adaptation schedule, we carried out experiments on 2048 × 2048 WorldView-3, WorldView-2 and GeoEye1 images, parts of larger tiles covering the cities of Adelaide (courtesy of DigitalGlobe©), Washington (courtesy of DigitalGlobe©) and Genoa (courtesy of DigitalGlobe©, provided by European Space Imaging), respectively. The image size (power of 2) simplifies the analysis of the proposed solution for obvious reasons but is by no means limiting in the validation of the basic idea. For each of the three cities/sensors, we disposed of four images—three for validation, the remaining for a test—as summarized in Table 2.

Table 2. Datasets for validation and test. Adelaide and Washington, courtesy of DigitalGlobe©. Genoa (DigitalGlobe©) provided by ESA.

Sensor	City	PAN Size	PAN GSD at Nadir	Spectral Bands	Validation Images	Test Images
GeoEye-1	Genoa	2048 × 2048	0.41 m	4	3	1
WorldView-2	Washington	2048 × 2048	0.46 m	8	3	1
WorldView-3	Adelaide	2048 × 2048	0.31 m	8	3	1

An RGB version of the MS component for each test image is shown in Figures 4–6, respectively. Overlaid on the images are the crops involved in the several adaptation phases

and some crops (A, B, C) that will be later recalled for the purpose of visual inspection of the results.

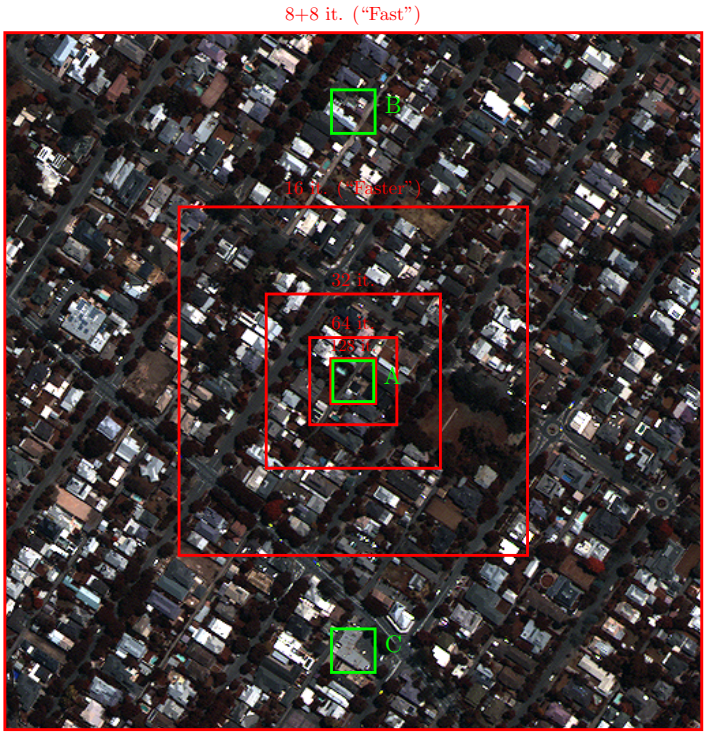


Figure 4. MS (RGB version) component of the WorldView-3 test image of Adelaide. Red concentric squared boxes indicate the crops used during the adaptation phases (the smaller the box, the earlier the phase). Green boxes (A, B, C) highlight three crops selected for visual inspection purposes.

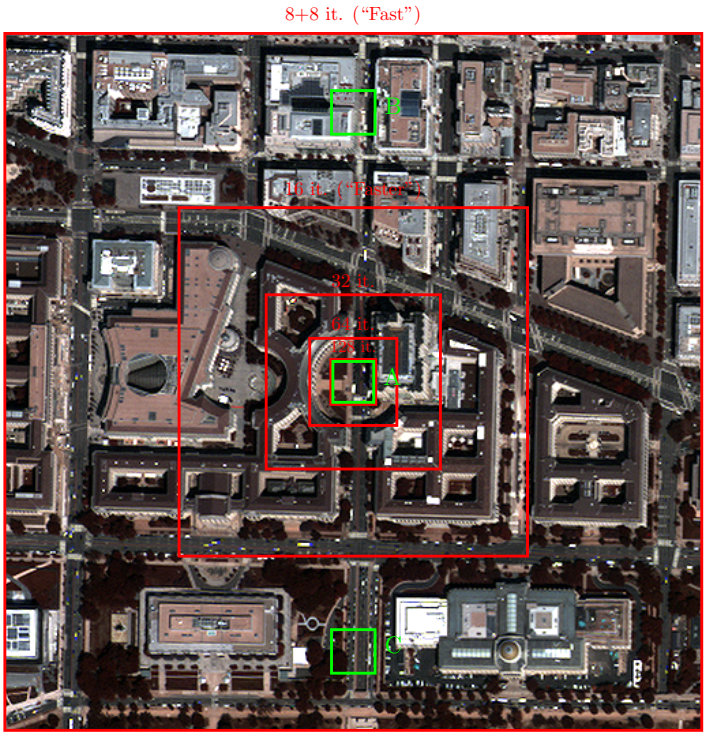


Figure 5. MS (RGB version) component of the WorldView-2 test image of Washington. Red concentric squared boxes indicate the crops used during the adaptation phases (the smaller the box, the earlier the phase). Green boxes (A, B, C) highlight three crops selected for visual inspection purposes.



Figure 6. MS (RGB version) component of the GeoEye-1 test image of Genoa. Red concentric squared boxes indicate the crops used during the adaptation phases (the smaller the box, the earlier the phase). Green boxes (A, B, C) highlight three crops selected for visual inspection purposes.

Table 3 summarizes the distribution of the number of iterations planned for each crop size and the corresponding computational time, by iteration and by phase. As it can be seen, the adaptation time for the proposed solution (see Fast Z-PNN on WV-$*$ to fix the ideas) was about 12 s against 40 or 100 s for the baseline scheme when using 100 (default choice) or 256 iterations, respectively. Moreover, it is worth noticing that most of the time (6.42 s) was spent by the last iterations on the full image. Similar considerations apply for GeoEye-1, as well as for the other two models. In particular, notice that for Z-DRPNN, all time scores scaled since this model is heavier (more parameters) compared to Z-PNN and Z-PanNet.

Table 3. Number of iterations for each image crop during the adaptation phase with corresponding unitary (Time/iter.) and cumulative (Time/phase) costs. The overall computational time for adaptation over WV-$*$/GE-1 images is about 12/6, 13/8 and 74/60 s for Z-PNN, Z-PanNet and Z-DRPNN, respectively, including an initial overhead of about a second for preliminary operations. Experiments have been run on a NVIDIA A100 GPU.

	Fast Z-PNN					
	crop size	128^2	256^2	512^2	1024^2	2048^2
	iterations	128	64	32	16	8 + 8
WV-2/-3	Time/iter. [ms]	8.1	9.5	29.8	104.5	401.2
	Time [s]	1.04	0.61	0.95	1.67	6.42
GE-1	Time/iter. [ms]	5.1	5.3	12.8	48.6	189.0
	Time [s]	0.66	0.34	0.41	0.78	3.02

Table 3. *Cont.*

	Fast Z-PanNet					
	crop size	128^2	256^2	512^2	1024^2	2048^2
	iterations	128	64	32	16	8 + 8
WV-2/-3	Time/iter. [ms]	11.1	15.2	32.8	118.1	426.2
	Time [s]	1.43	0.97	1.05	1.89	6.82
GE-1	Time/iter. [ms]	12.2	13.0	13.7	52.4	210.3
	Time [s]	1.56	0.83	0.43	0.84	3.36
	Fast Z-DRPNN					
	crop size	128^2	256^2	512^2	1024^2	2048^2
	iterations	128	64	32	16	8 + 8
WV-2/-3	Time/iter. [ms]	26.8	30.2	165.6	872.5	3060.8
	Time [s]	3.42	1.93	5.30	13.96	48.97
GE-1	Time/iter. [ms]	26.3	28.8	143.9	510.4	2620.3
	Time [s]	3.36	1.84	4.60	8.16	41.92

Actually, from a more careful inspection of Table 3, it can be noticed that the time per iteration tended to increase by a factor of 4, moving from one crop size to the next one when these were larger. In fact, Fast Z-PNN obtained the following time multipliers on WV-$*$: 1.17 ($128^2 \rightarrow 256^2$), 3.14, 3.51 and 3.84 ($1024^2 \rightarrow 2048^2$). This should not be surprising since the crop area increased by 4 when doubling its linear dimensions and also because the computational time on parallel computing units, such as GPUs, does not always scale linearly with the image size, particularly when the input images are relatively small or too large, causing memory swaps. Assuming to be in a linear regime where the iteration cost grows linearly with the image size (area), considering that the number of iterations halves from one phase to the next one, the time consumption per phase eventually doubled, moving from a phase to the next one. Asymptotically, each new phase took a time comparable to the time accumulated by all previous phases. Consequently, by skipping the last phase, one would save approximately half of the computational burden, paying something in terms of accuracy.

Based on the above considerations, we also tested the lighter configuration, "Fast*er*", of the proposed method, where we skipped the last tuning phase with an early stop, as indicated in Table 1.

The experimental evaluation was split in two parts. On one side, the proposed Fast and Faster variants were compared to the corresponding baselines (Z-$*$) directly in terms of loss achieved during the target-adaptation phase. This was completed separately for both spectral and spatial loss components and using the validation images only. The results of this analysis for the whole validation dataset are summarized in Tables 4 and 5, while Figure 7 displays the loss curves for a sample image. Moreover, Figure 8 compares the different target-adaptive schemes for Z-PNN using some sample visual results.

Table 4. Run time for target-adaptation. T' and T'' indicate the run time for Faster and Fast Z-PNN, who achieve the loss values $\mathcal{L}_*^{\text{Faster}}$ and $\mathcal{L}_*^{\text{Fast}}$, respectively ($\mathcal{L}_*^0$: pretrained model). T'_Z and T''_Z are the run time for Z-PNN to reach the same loss levels achieved by its Faster and Fast versions, respectively. All times are given in seconds. In (a) is the WorldView-$*$ case, in (b) GeoEye-1. Initialization time (about a second) fixed for all is not counted. Bold numbers indicate cases with non decreasing spatial loss.

Image	$\mathcal{L}_\lambda^0$ (0[s])	$\mathcal{L}_\lambda^{\text{Faster}}$ (T'=4.3[s])	$\mathcal{L}_\lambda^{\text{Fast}}$ (T''=10.7[s])	T'_Z	T''_Z	T'_Z/T' (gain)	T''_Z/T'' (gain)
WV-2 #1	0.008	0.006	0.006	49.9	61.5	11.6	5.75
WV-2 #2	0.009	0.006	0.005	85.2	103	19.8	9.62
WV-2 #3	0.013	0.009	0.009	76.4	84.8	17.8	7.93
WV-3 #1	0.019	0.015	0.015	79.0	87.2	18.4	8.14
WV-3 #2	0.018	0.013	0.013	142	171	33.1	16.2
WV-3 #3	0.014	0.011	0.010	124	150	28.9	14.9
Image	$\mathcal{L}_S^0$ (0[s])	$\mathcal{L}_S^{\text{Faster}}$ (T'=4.3[s])	$\mathcal{L}_S^{\text{Fast}}$ (T''=10.7[s])	T'_Z	T''_Z	T'_Z/T' (gain)	T''_Z/T'' (gain)
WV-2 #1	0.044	**0.046**	**0.047**	n.a.	n.a.	n.a.	n.a.
WV-2 #2	0.028	**0.050**	**0.049**	n.a.	n.a.	n.a.	n.a.
WV-2 #3	0.045	**0.057**	**0.056**	n.a.	n.a.	n.a.	n.a.
WV-3 #1	0.154	0.099	0.097	88.0	116.3	20.5	10.9
WV-3 #2	0.146	0.099	0.092	47.0	76.4	10.9	7.1
WV-3 #3	0.121	0.079	0.078	56.2	76.4	13.1	7.1

(a)

Image	$\mathcal{L}_\lambda^0$ (0[s])	$\mathcal{L}_\lambda^{\text{Faster}}$ (T'=2.9[s])	$\mathcal{L}_\lambda^{\text{Fast}}$ (T''=4.7[s])	T'_Z	T''_Z	T'_Z/T' (gain)	T''_Z/T'' (gain)
GE-1 #1	0.011	0.010	0.010	75.5	95.1	26.0	20.2
GE-1 #2	0.016	0.013	0.013	45.5	56.4	15.7	12.0
GE-1 #3	0.013	0.011	0.011	57.0	68.9	19.7	14.7
Image	$\mathcal{L}_S^0$ (0[s])	$\mathcal{L}_S^{\text{Faster}}$ (T'=2.9[s])	$\mathcal{L}_S^{\text{Fast}}$ (T''=4.7[s])	T'_Z	T''_Z	T'_Z/T' (gain)	T''_Z/T'' (gain)
GE-1 #1	0.130	0.120	0.115	38.9	52.2	13.4	11.1
GE-1 #2	0.137	0.135	0.131	54.0	69.4	18.6	14.8
GE-1 #3	0.118	**0.118**	0.114	n.a	30.6	n.a.	6.6

(b)

Table 5. Avarage run time for target-adaptation for all involved models, referred to the spectral losses.

Model	**Sensor**	T'_Z	T''_Z	T'_Z/T'	T''_Z/T''
Z-PNN	WV-2	70.5	83.1	16.4	7.77
Z-PNN	WV-3	115	136	26.8	13.1
Z-PNN	GE-1	59.3	73.5	20.5	15.6
Z-PanNet	WV-2	81.2	91.1	18.9	8.51
Z-PanNet	WV-3	97.3	111	22.6	10.4
Z-PanNet	GE-1	41.7	46.1	14.4	9.80
Z-DRPNN	WV-2	579	665	19.9	9.2
Z-DRPNN	WV-3	184	292	6.33	4.03
Z-DRPNN	GE-1	221	264	16.2	7.77

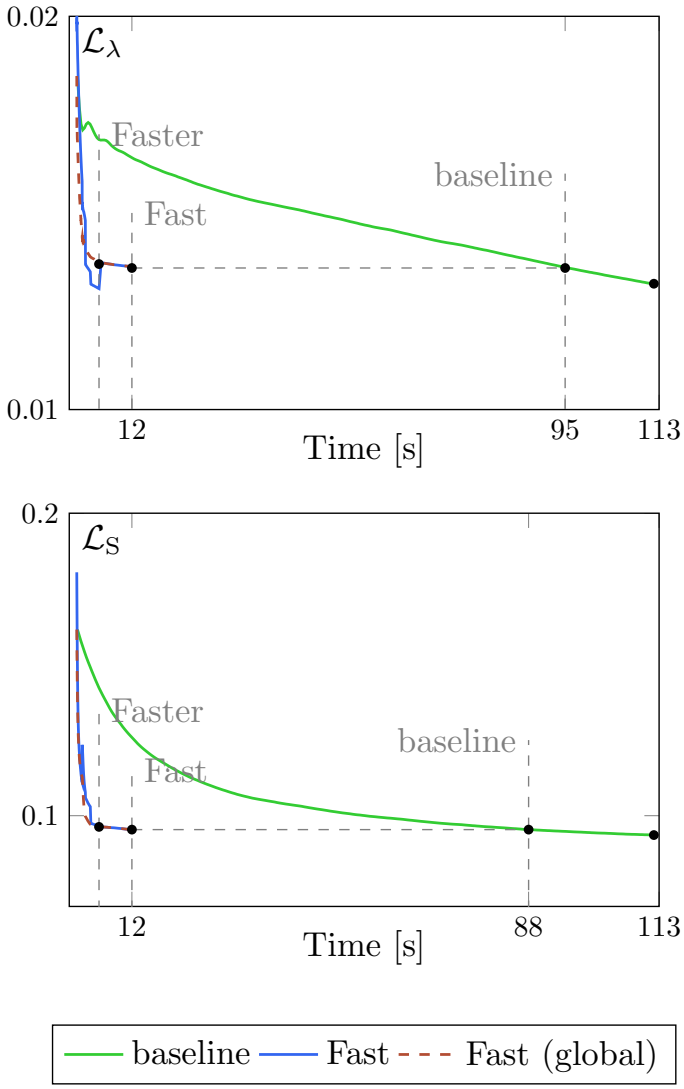


Figure 7. Loss decay during the adaptation phase for Z-PNN on a sample WV-3 image: spectral (**top**) and spatial (**bottom**) loss terms. The loss of the baseline model is shown in green; the loss of the proposed solution (computed on the current crop) is shown in blue; dashed red lines show the loss of the proposed model during training computed on the whole image.

On the other hand, the test images were used for a comparative quality assessment in terms of both pansharpening numerical indexes and subjective visual inspection. For a more robust quality evaluation, we resorted to the pansharpening benchmark toolbox [11], which provides the implementation of several state-of-the-art methods and several quality indexes, e.g., the spectral $D_\lambda^{(K)}$ and spatial D_S consistency indexes. We integrated the benchmark with the Machine Learning toolbox proposed in [39], which provides additional CNN-based methods. Furthermore, the evaluation was carried out using the additional indexes D_ρ, R-SAM, R-ERGAS and R-$Q2^n$, recently proposed in [40]. All comparative methods are summarized in Table 6. The results are gathered in Tables 7–9, whereas sample visual results are given in Figures 9–11. A deeper discussion about these results is left to Section 4.

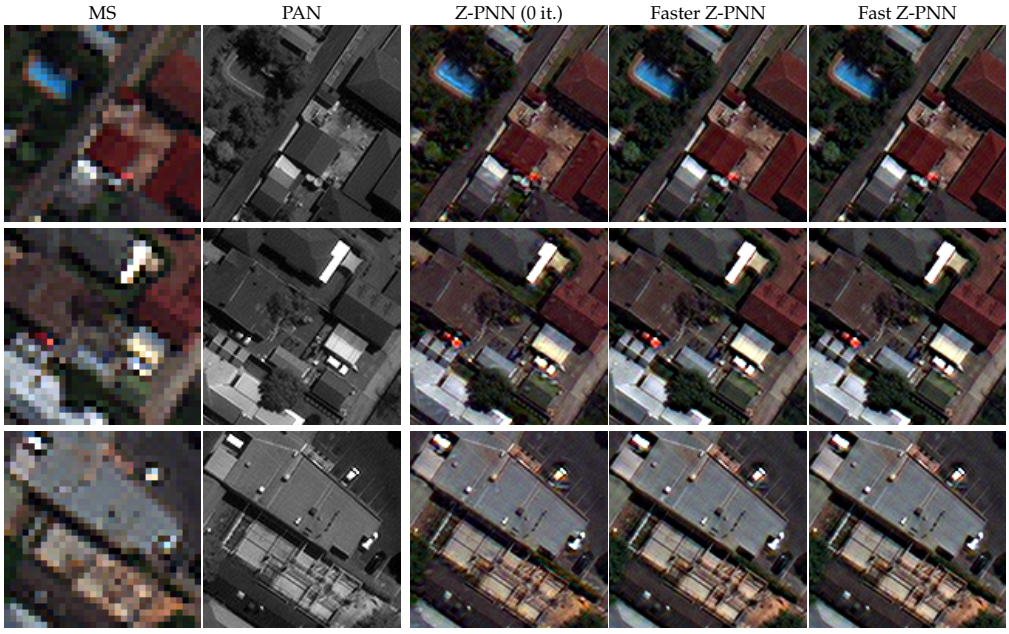


Figure 8. Pansharpening results without adaptation (0 it.), with partial (Faster) and full (Fast) adaptation according to the proposed scheme. From top to bottom: WV-3 Adelaide crops A, B and C, respectively.

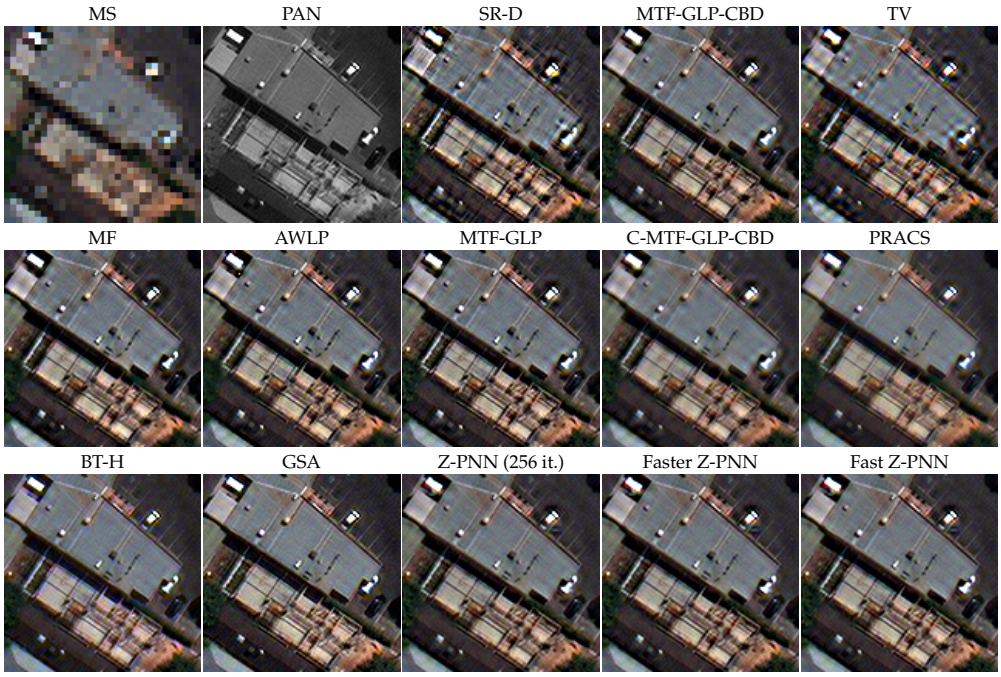


Figure 9. Pansharpening results on the WorldView-3 Adelaide image (crop C).

Table 6. Detailed list of all reference methods.

Component Substitution (CS)
BT-H [41], BDSD [42], C-BDSD [43], BDSD-PC [44], GS [15], GSA [45], C-GSA [19], PRACS [46]
Multiresolution Analysis (MRA)
AWLP [47], MTF-GLP [48], MTF-GLP-FS [49], MTF-GLP-HPM [48], MTF-GLP-HPM-H [41], MTF-GLP-HPM-R [50], MTF-GLP-CBD [51], C-MTF-GLP-CBD [19], MF [52]
Variational Optimization (VO)
FE-HPM [22], SR-D [29], TV [20]
Machine Learning (ML)
BDPN [37], DiCNN [53], DRPNN [30], FusionNet [54], MSDCNN [27], PanNet [25], PNN [24], A-PNN [36], A-PNN-TA [36], Z-PNN [35]

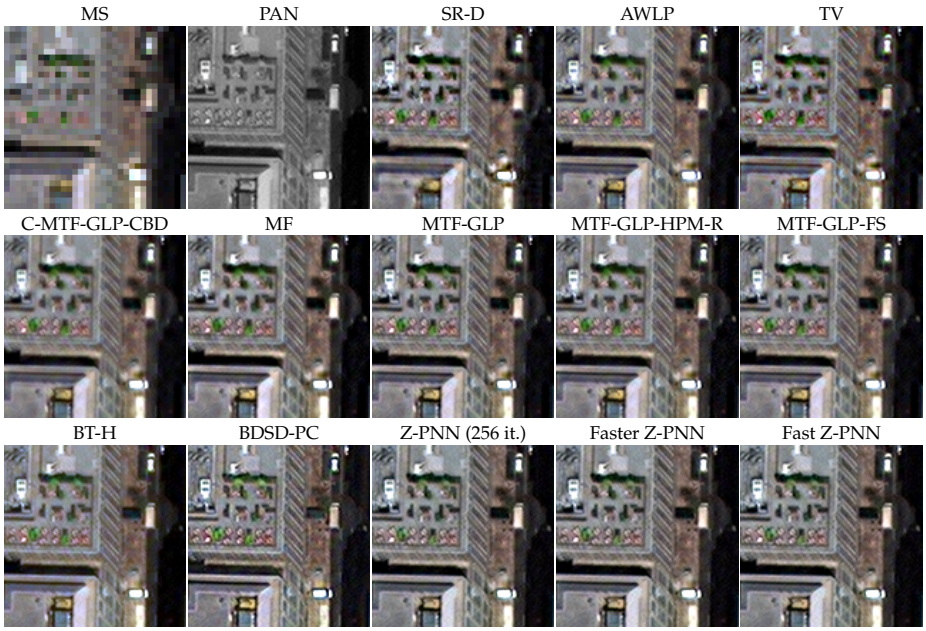


Figure 10. Pansharpening results on the WorldView-2 Washington image (crop B).

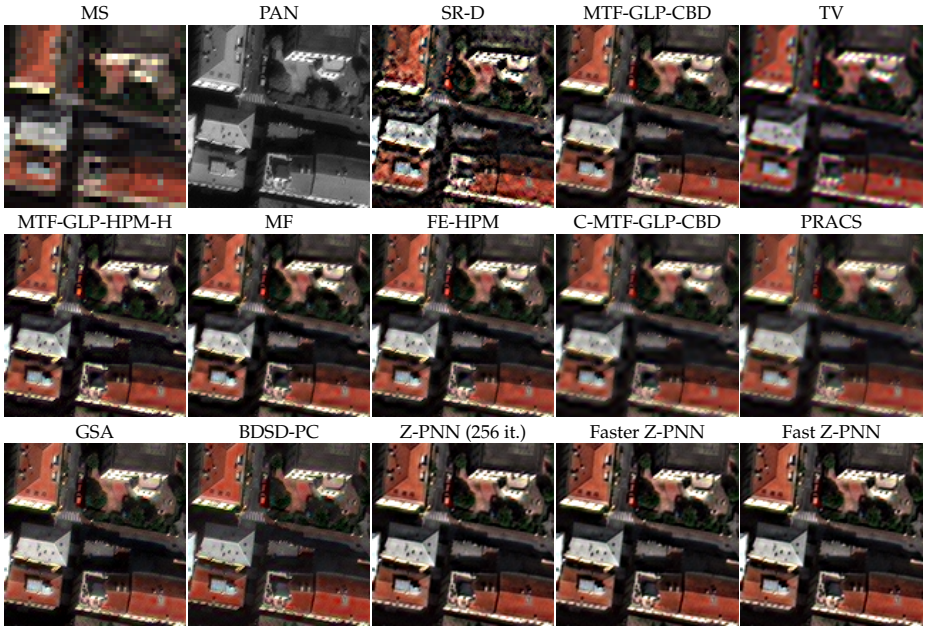


Figure 11. Pansharpening results on the GeoEye-1 Washington image (crop B).

Table 7. Numerical results on the 2048 × 2048 WorldView-3 test image (Adelaide), courtesy of DigitalGlobe©.

Method (Ideal)	$D_\lambda^{(K)}$ **(0.0000)**	**R-Q2^n (1.0000)**	**R-SAM (0.0000)**	**R-ERGAS (0.0000)**	D_S **(0.0000)**	D_ρ **(0.0000)**
EXP	0.0449	0.9315	5.5175	4.7394	0.1585	0.8557
BT-H	0.1216	0.9416	5.4830	4.3314	0.1283	0.0702
BDSD	0.1502	0.9259	6.4563	4.9037	0.1143	0.0971
C-BDSD	0.1349	0.9247	6.6180	5.0875	0.0775	0.1512
BDSD-PC	0.1484	0.9281	6.3480	4.8047	0.1211	0.0836
GS	0.1717	0.9165	6.0261	5.1157	0.1097	0.0974
GSA	0.1291	0.9433	5.9260	4.4414	0.1133	0.0738
C-GSA	0.1256	0.9415	6.2609	4.5317	0.0899	0.1748
PRACS	0.0688	0.9454	5.4757	4.2258	0.0665	0.1944
AWLP	0.0320	0.9536	5.6340	3.9604	0.0728	0.0907
MTF-GLP	0.0354	0.9534	5.6007	3.9853	0.1067	0.0723
MTF-GLP-FS	0.0344	0.9509	5.5752	4.0743	0.0973	0.1071
MTF-GLP-HPM	0.0423	0.9528	5.7651	4.0230	0.0900	0.0817
MTF-GLP-HPM-H	0.0366	0.9530	5.6177	4.0437	0.1163	0.0799
MTF-GLP-HPM-R	0.0362	0.9512	5.6276	4.0770	0.0848	0.1230
MTF-GLP-CBD	0.0343	0.9506	5.5743	4.0853	0.0954	0.1115
C-MTF-GLP-CBD	0.0348	0.9458	5.5883	4.2801	0.0522	0.2480
MF	0.0439	0.9588	5.5940	3.7625	0.0819	0.0972
FE-HPM	0.0344	0.9515	5.6433	4.1464	0.0810	0.0983
SR-D	0.0183	0.9466	5.6796	4.4005	0.0264	0.2801
TV	0.0373	0.9699	4.2437	3.2592	0.0529	0.2285
BDPN	0.2144	0.8822	6.1821	5.8480	0.1064	0.1882
DiCNN	0.1862	0.8909	6.4608	5.9225	0.0684	0.4236
DRPNN	0.2015	0.8880	6.1327	5.7324	0.0867	0.2117
FusionNet	0.1056	0.9322	5.7242	4.7337	0.0583	0.4142
MSDCNN	0.2059	0.8691	6.3453	5.9449	0.1045	0.2288
PanNet	0.0352	0.9493	5.4097	4.1781	0.0364	0.3466
PNN	0.1839	0.8861	7.7183	7.0513	0.0523	0.4755
A-PNN	0.0736	0.9429	5.4923	4.3273	0.0545	0.5640
A-PNN-FT	0.0543	0.9488	5.3547	4.1822	0.0318	0.3154
Z-DRPNN (0 it.)	0.0742	0.9451	7.0813	4.7282	0.0949	0.0543
Z-DRPNN(100 it.)	0.0316	0.9463	5.7685	4.3278	0.0601	0.0753
Z-DRPNN (256 it.)	0.0307	0.9467	5.7051	4.3388	0.0625	0.0797
Faster Z-DRPNN (early stop)	0.0278	0.9466	5.7127	4.3310	0.0700	0.0788
Fast Z-DRPNN	0.0270	0.9470	5.6767	4.3086	0.0670	0.0777
Z-PanNet (0 it.)	0.0378	0.9490	5.5755	4.1746	0.0731	0.1634
Z-PanNet (100 it.)	0.0391	0.9514	5.6135	4.0910	0.0911	0.1288
Z-PanNet (256 it.)	0.0376	0.9508	5.6596	4.1202	0.0915	0.1059
Faster Z-PanNet (early stop)	0.0370	0.9499	5.6696	4.1502	0.0895	0.1111
Fast Z-PanNet	0.0370	0.9501	5.6698	4.1443	0.0898	0.1087
Z-PNN (0 it.)	0.0859	0.9385	6.2438	4.3860	0.0789	0.1634
Z-PNN (100 it.)	0.0599	0.9466	5.9572	4.1688	0.0922	0.1048
Z-PNN (256 it.)	0.0458	0.9494	5.7066	4.1088	0.0941	0.0959
Faster Z-PNN (early stop)	0.0496	0.9476	5.7986	4.1680	0.0943	0.0985
Fast Z-PNN	0.0497	0.9475	5.8013	4.1693	0.0943	0.0985

Table 8. Numerical results on the 2048 × 2048 WorldView-2 test image (Washington), courtesy of DigitalGlobe©.

Method (Ideal)	$D_\lambda^{(K)}$ **(0.0000)**	**R-Q2^n (1.0000)**	**R-SAM (0.0000)**	**R-ERGAS (0.0000)**	D_S **(0.0000)**	D_ρ **(0.0000)**
EXP	0.0316	0.9633	2.7902	2.5396	0.2066	0.7918
BT-H	0.0897	0.9623	2.8052	2.4942	0.1278	0.0670
BDSD	0.1651	0.9229	4.1417	3.5450	0.0938	0.1198
C-BDSD	0.1590	0.9223	4.1914	3.7182	0.0738	0.1594
BDSD-PC	0.1335	0.9425	3.2018	3.0935	0.1293	0.0698
GS	0.1207	0.9454	3.3927	3.0398	0.1151	0.0797
GSA	0.0860	0.9651	3.0587	2.5272	0.1037	0.0657
C-GSA	0.0871	0.9646	3.0732	2.5413	0.1106	0.0715
PRACS	0.0489	0.9676	2.8387	2.3887	0.0827	0.1766
AWLP	0.0192	0.9740	2.8182	2.1870	0.0668	0.0967
MTF-GLP	0.0226	0.9739	2.7728	2.1925	0.0748	0.0790
MTF-GLP-FS	0.0218	0.9734	2.7755	2.2084	0.0695	0.0983
MTF-GLP-HPM	0.0232	0.9739	2.7936	2.1962	0.0728	0.0778
MTF-GLP-HPM-H	0.0254	0.9731	2.8329	2.2148	0.0904	0.0868
MTF-GLP-HPM-R	0.0222	0.9734	2.8222	2.2180	0.0651	0.1057
MTF-GLP-CBD	0.0217	0.9733	2.7761	2.2104	0.0685	0.1003
C-MTF-GLP-CBD	0.0212	0.9722	2.7745	2.2544	0.0582	0.1404
MF	0.0306	0.9782	2.7859	2.0214	0.0520	0.0954
FE-HPM	0.0236	0.9737	2.8319	2.2381	0.0583	0.0929
SR-D	0.0104	0.9720	2.7168	2.3090	0.0329	0.2567
TV	0.0336	0.9877	1.7697	1.4760	0.0475	0.2043
BDPN	0.0913	0.9567	3.0843	2.6526	0.0863	0.1388
DiCNN	0.0806	0.9603	2.8935	2.5543	0.0446	0.2330
DRPNN	0.0620	0.9670	3.0265	2.3686	0.0611	0.1604
FusionNet	0.0546	0.9689	2.6794	2.2979	0.0461	0.2105
MSDCNN	0.0709	0.9633	3.1426	2.4472	0.0718	0.1415
PanNet	0.0224	0.9733	2.7059	2.2169	0.0402	0.2714
PNN	0.0504	0.9666	2.9938	2.4126	0.0740	0.1839
A-PNN	0.0374	0.9704	2.6580	2.3055	0.0662	0.2806
A-PNN-FT	0.0386	0.9704	2.6773	2.2939	0.0511	0.1846
Z-DRPNN (0 it.)	0.1241	0.9437	7.2789	4.5430	0.0616	0.0558
Z-DRPNN(100 it.)	0.0283	0.9723	3.0126	2.2586	0.0413	0.0510
Z-DRPNN (256 it.)	0.0231	0.9729	2.8822	2.2338	0.0484	0.0544
Faster Z-DRPNN (early stop)	0.0208	0.9723	2.9352	2.2711	0.0498	0.0613
Fast Z-DRPNN	0.0204	0.9728	2.8707	2.2428	0.0510	0.0596
Z-PanNet (0 it.)	0.0273	0.9758	2.8358	2.1133	0.0424	0.1628
Z-PanNet (100 it.)	0.0300	0.9759	2.8275	2.1095	0.0471	0.1404
Z-PanNet (256 it.)	0.0297	0.9755	2.8169	2.1246	0.0493	0.1177
Faster Z-PanNet (early stop)	0.0309	0.9757	2.8136	2.1069	0.0513	0.1187
Fast Z-PanNet	0.0306	0.9757	2.8161	2.1095	0.0508	0.1167
Z-PNN (0 it.)	0.0738	0.9603	3.3773	2.5244	0.1101	0.0367
Z-PNN (100 it.)	0.0437	0.9689	3.1370	2.3212	0.0892	0.0398
Z-PNN (256 it.)	0.0361	0.9713	2.9781	2.2576	0.0828	0.0437
Faster Z-PNN (early stop)	0.0384	0.9699	3.0618	2.3141	0.0845	0.0432
Fast Z-PNN	0.0398	0.9699	3.0705	2.3012	0.0869	0.0419

Table 9. Numerical results on the 2048 × 2048 GeoEye-1 test image (Genoa), © DigitalGlobe, Inc. (2018), provided by European Space Agency.

Method (Ideal)	$D_\lambda^{(K)}$ **(0.0000)**	**R-Q2^n (1.0000)**	**R-SAM (0.0000)**	**R-ERGAS (0.0000)**	D_S **(0.0000)**	D_ρ **(0.0000)**
EXP	0.1321	0.8515	4.1849	6.0561	0.1103	0.8244
BT-H	0.3213	0.8181	6.1603	6.0892	0.0703	0.0818
BDSD	0.2807	0.8470	4.5871	5.7298	0.1283	0.0749
C-BDSD	0.2472	0.8505	4.7650	5.7536	0.1065	0.1032
BDSD-PC	0.2845	0.8450	4.5667	5.7369	0.1305	0.0742
GS	0.2955	0.8262	4.6896	5.8848	0.1274	0.0859
GSA	0.2979	0.8213	5.9619	5.7529	0.1086	0.0525
C-GSA	0.2897	0.8309	6.1918	5.7073	0.0965	0.1087
PRACS	0.1718	0.8657	4.4010	5.5338	0.0453	0.2564
AWLP	0.1069	0.8862	4.1538	5.2522	0.0799	0.1090
MTF-GLP	0.1104	0.8924	4.0992	5.0684	0.0815	0.0913
MTF-GLP-FS	0.1106	0.8887	4.0977	5.1601	0.0855	0.1079
MTF-GLP-HPM	0.1135	0.8944	4.0271	5.0204	0.0712	0.0876
MTF-GLP-HPM-H	0.1162	0.8952	3.9776	5.0131	0.0468	0.0910
MTF-GLP-HPM-R	0.1109	0.8903	4.0932	5.1229	0.0776	0.1141
MTF-GLP-CBD	0.1108	0.8875	4.0960	5.1896	0.0857	0.1129
C-MTF-GLP-CBD	0.1175	0.8743	4.1409	5.5269	0.0343	0.3105
MF	0.1240	0.8948	4.0178	4.8697	0.0795	0.1031
FE-HPM	0.1175	0.8943	4.0406	4.9831	0.0849	0.0996
SR-D	0.0697	0.8965	4.2034	5.3477	0.0143	0.3989
TV	0.1643	0.8907	3.6706	4.9564	0.0392	0.7326
PNN	0.1126	0.8834	3.6556	5.3172	0.0199	0.4615
A-PNN	0.1026	0.8841	3.7173	5.3493	0.0371	0.5225
A-PNN-FT	0.1161	0.8782	3.7671	5.4407	0.0167	0.3543
Z-DRPNN (0 it.)	0.1210	0.8903	3.8684	5.0001	0.0355	0.1370
Z-DRPNN(100 it.)	0.0786	0.9022	3.7652	5.1102	0.0838	0.1319
Z-DRPNN (256 it.)	0.0771	0.9041	3.8007	5.1487	0.1031	0.1323
Faster Z-DRPNN (early stop)	0.0965	0.8939	3.8258	5.1537	0.0986	0.1376
Fast Z-DRPNN	0.1036	0.8909	3.8484	5.1589	0.0968	0.1369
Z-PanNet (0 it.)	0.1196	0.8917	4.0194	5.1717	0.1020	0.1224
Z-PanNet (100 it.)	0.1174	0.8933	3.9482	5.0904	0.0732	0.1122
Z-PanNet (256 it.)	0.1094	0.8938	3.9999	5.0956	0.0448	0.1040
Faster Z-PanNet (early stop)	0.1051	0.8933	4.1054	5.1078	0.0372	0.1105
Fast Z-PanNet	0.1006	0.8939	4.0830	5.0895	0.0346	0.1100
Z-PNN (0 it.)	0.1710	0.8804	4.0835	5.1055	0.0991	0.1494
Z-PNN (100 it.)	0.1482	0.8855	3.9509	5.0134	0.0913	0.1360
Z-PNN (256 it.)	0.1454	0.8860	3.9298	5.0046	0.0829	0.1209
Faster Z-PNN (early stop)	0.1339	0.8922	3.9097	4.9436	0.0787	0.1427
Fast Z-PNN	0.1373	0.8896	3.9117	4.9869	0.0777	0.1389

4. Discussion

Let us now analyze in depth the provided results, starting from a comparison with the baseline models Z-∗.

In Figure 7, we show the loss decay during the adaptation phase, separately for the spectral (top) and spatial (bottom) terms, as a function of the running time rather than the iteration count. This experiment refers to a WorldView-3 image processed by Z-PNN models, but similar behaviors have been observed for the other images, regardless of the employed base model. The loss terms for the baseline (Z-PNN) and the proposed Fast Z-PNN are plotted in green and blue, respectively. Although the loss is a per-pixel loss (spatial average), so that these curves are dimensionally consistent, the latter refers to the current crop while the former is an average on the whole image. For this reason, for the proposed (Fast/Faster) solution, we also computed the value of the loss on the whole image

at each iteration (red dashed line). Surprisingly, the "global" loss tightly follows the "local" (to crop) loss, showing a more regular decay, thanks to a wider average. These plots clearly show the considerable computational gain achievable for any fixed target loss level. It is worth noticing, for example, that the loss levels achieved by the proposed Fast scheme in 12 s (256 iterations) are reached by the baseline approach in about 90 s (∼200 iterations on the full image). Besides, the Faster version reaches almost the same loss levels of the Fast version in about 5.7 s.

In Table 4, for each validation image, we quantify the computational gains achieved by Faster and Fast Z-PNN in terms of time consumption. For each image, it is indicated, separately for the spectral (top table) and spatial (bottom table) loss components, the value without adaptation ($\mathcal{L}_*^0$) and those achieved by Faster ($\mathcal{L}_*^{\text{Faster}}$) and Fast ($\mathcal{L}_*^{\text{Fast}}$) adaptation schemes (these values refer to the loss assessed on the whole image, not to the one computed on the crops for gradient descent), whose run times (T' and T'') do not depend on the specific image (they drop when working with GeoEye-1 instead of WV-$*$ because of the smaller number of spectral bands). Then, we report the times T'_Z and T''_Z needed for Z-PNN to reach the loss levels $\mathcal{L}_*^{\text{Faster}}$ and $\mathcal{L}_*^{\text{Faster}}$, respectively. We can fairly read T'_Z and T''_Z as the times needed for Z-PNN to achieve the same spectral (top table) or spatial (bottom table) target levels of its faster versions. Consequently, T'_Z/T' and T''_Z/T'' represent the computational gains of the proposed solution. Similar results, not shown for brevity, have been registered for Z-PanNet and Z-DRPNN. It is worth noticing that whereas for the spectral loss the gain is always (well) larger than 1, for the spatial component, there can occur cases where it is already pretty low (see $\mathcal{L}_S^0$ in Table 4); hence, only the spectral loss decays while the spatial one either keeps constant or even grows a little (see bold numbers in Table 4). In these cases, the gains cannot be computed, but actually, it make sense to focus on the spectral behavior only, as no adaptation is needed on the spatial side.

In general, from all experiments that we carried out, it emerges that the adaptation is mostly needed for the reduction of the spectral distortion rather than for the spatial one. This is a particular feature of the Z-$*$ framework, which leverages a spatial consistency loss based on correlation (Equation (4)) that shows a quite robust behavior. On the basis of the above reasons, it makes sense to focus on the spectral loss only for checking the quality alignment between Z-$*$ and the proposed variants. Therefore, we can give a look to Table 5, which provides the average gains for all models, averaged by sensor, but limited to the spectral part. Overall, it can be observed that the Fast models allow for obtaining gains ranging from 4.03 to 15.6 (9.6 on average), whereas Faster models provide gains between 6.33 to 26.8 (17.9 on average), at the price of a little increase of the loss components (compare the loss levels in Table 4).

Let us now move to the analysis of the results obtained on the test images for a comparison with the state-of-the-art solutions. Starting from the numerical results gathered in Tables 7 (WV-3), 8 (WV-2) and 9 (GE-1), we can observe that the most important achievement is that the proposed Faster and Fast solutions are exactly where they are expected to be, i.e., between Z-$*$ (100 it.) and Z-$*$ (256 it.) almost always, coherently with the loss levels shown in Figure 7 and Table 4, with few exceptions. In some cases, the proposals behave even better than the baseline (e.g., Fast-Z-PNN against Z-PNN on GeoEye-1, Table 9). Moreover, it is worth noticing that Faster Z-$*$ tightly follows Fast Z-$*$ and Z-$*$ (256 it.), confirming our initial guess that the need for tuning is mostly due to geometric or atmospheric misalignments between the training and testing datasets rather than grounding content mismatches.

Concerning the overall comparison, it must be underlined that all indexes are used on the full-resolution image without any ground-truth. They assess the consistency rather than the synthesis properties, and each reflects some arbitrary assumption. Both D_S and D_ρ deal with the spatial consistency, while the remaining relate to the spectral consistency. While the latter group shows a good level of agreement, the former looks much less correlated. For a deeper discussion on this issue that is a little out of scope here, the Reader is referred

to [35]. In addition, the goal of the present contribution is on efficiency rather than on accuracy; therefore, we leave these results to the reader without further discussion.

Beside numerical assessment, the visual inspection of the results is a fundamental complementary check to gain insight into the behavior of the compared methods. Let us first analyze the impact of the tuning phase by comparing the pretrained model Z-$*$ (0 it.) with the proposed target-adaptive solutions, Faster and Fast versions. Figure 8 shows a few crops from WV-3 Adelaide image (Figure 4), with the related pansharpened images obtained with the Z-PNN variants. Remarkable differences can be noticed between the pretrained model and the two target-adaptive options. The Reader may notice the visible artifacts introduced by the pretrained model occurring, for example, in the pool of crop A or on several building roofs (spotted artifacts), which are removed by the proposed solutions, thanks to the tuning. On the other hand, there is no noticeable difference between the Faster and Fast configurations proposed. It is also worth noticing that this alignment between the Fast and the Faster configurations hold indistinguishably on crop A (always involved in the tuning) and crops B and C that are not involved in the tuning process in the Faster version (see Figure 4). Similar considerations can be noted from the experiments, not shown for brevity, carried out on the other datasets and/or using different baseline models.

Figures 9–11 show, again, on some selected crops, the pansharpening results obtained by the proposed solutions and by the best performing ones among all comparative methods listed in Table 6 for the WorldView-3, WorldView-2 and GeoEye-1 images, respectively. In particular, these results confirm the most relevant observation that Fast Z-PNN, Faster Z-PNN and the baseline Z-PNN are nearly indistinguishable, in line with the numerical results discussed above, further underlining that the registered computational gain has been achieved without sacrificing accuracy. Regarding the comparison with the other methods, from the visual point of view, the proposed solutions are aligned with the best ones, without appreciable spectral or spatial distortion and a very high contrast level.

Finally, in order to further prove the robustness of the proposed approach, we ran a cross-image experiment. Taking two different WV-3 sample images whose content is sufficiently different (see Figure 12), we ran Fast Z-PNN on the first one (a), which plays as a target image. On the same image, we also tested Z-PNN without adaptation (0 it.) and the model adapted on image (b) using the Fast configuration. The numerical results, shown in (c), provide further confirmation that the actual content of the image has a minor impact on pansharpening, in the tuning phase, with respect to acquisition geometry and atmospheric conditions.



(**a**) (**b**)

Numerical results for the target image (a)

Method	$D_\lambda^{(K)}$	R-Q2^n	R-SAM	R-ERGAS	D_S	D_ρ
Z-PNN (0 it.)	0.0783	0.9318	6.3607	4.5391	0.0723	0.1566
Fast Z-PNN	0.0498	0.9348	6.0955	4.5345	0.0868	0.0996
Fast Z-PNN (b)	0.0481	0.9309	6.1919	4.6722	0.0781	0.1130

(**c**)

Figure 12. Cross-image adaptation test. (**a**) target image; (**b**) a different image from the same validation dataset (WV-3); (**c**) numerical results. Fast Z-PNN (**b**) corresponds to the results on (**a**) with the model fitted on (**b**).

5. Conclusions

In this work, we have presented a new target-adaptive framework for CNN pansharpening. A full-version named *Fast* and a quicker configuration named *Faster* have been tested over the recently proposed unsupervised full-resolution models: Z-PNN, Z-PanNet and Z-DRPNN [35]. The results clearly show that the inference time reduces by a factor of 10 or more, which doubles for the Faster version, without sacrificing accuracy. Such gain has been achieved thanks to the assumption, experimentally validated here, that generalization problems occur when training and test datasets come from different acquisitions, hence because of geometrical and/or atmospheric changes, whereas different areas of a same image can be considered sufficiently "homogeneous" from the learning point of view. Starting from this consideration, the proposed target adaptive scheme does not make use of the full test image for its unsupervised adaptation from the beginning, but, starting from a central portion, it progressively encloses all data in the process. Indeed, the Faster version only uses 25% of the image, obtaining nearly the same quality level.

Future work may explore different spatial sampling strategies, such as adaptive or dynamic selection of the tuning crops, in order to further improve the robustness and generality of the proposed framework.

Author Contributions: Conceptualization, G.S. and M.C.; methodology, G.S. and M.C.; software, M.C.; validation, G.S. and M.C.; formal analysis, G.S. and M.C.; investigation, G.S. and M.C.; resources, G.S. and M.C.; data curation, M.C.; writing—original draft preparation, G.S.; writing—review and editing, G.S. and M.C.; visualization, M.C.; supervision, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank DigitalGlobe© and ESA for providing sample images for research purposes.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moser, G.; Serpico, S. Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2972–2982. [CrossRef]
2. Chen, Y.; Bruzzone, L. Self-Supervised SAR-Optical Data Fusion of Sentinel-1/-2 Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5406011. [CrossRef]
3. Errico, A.; Angelino, C.V.; Cicala, L.; Podobinski, D.P.; Persechino, G.; Ferrara, C.; Lega, M.; Vallario, A.; Parente, C.; Masi, G.; et al. SAR/multispectral image fusion for the detection of environmental hazards with a GIS. In Proceedings of the SPIE—The International Society for Optical Engineering, Amsterdam, The Netherlands, 22–25 September 2014; Volume 9245.
4. Scarpa, G.; Gargiulo, M.; Mazza, A.; Gaetano, R. A CNN-Based Fusion Method for Feature Extraction from Sentinel Data. *Remote Sens.* **2018**, *10*, 236. [CrossRef]
5. Vivone, G.; Alparone, L.; Chanussot, J.; Mura, M.D.; Garzelli, A.; Licciardi, G.A.; Restaino, R.; Wald, L. A Critical Comparison Among Pansharpening Algorithms. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2565–2586. [CrossRef]
6. Gargiulo, M.; Mazza, A.; Gaetano, R.; Ruello, G.; Scarpa, G. A CNN-Based Fusion Method for Super-Resolution of Sentinel-2 Data. In Proceedings of the IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 4713–4716. [CrossRef]
7. Ciotola, M.; Ragosta, M.; Poggi, G.; Scarpa, G. A full-resolution training framework for Sentinel-2 image fusion. In Proceedings of the IGARSS 2021, Brussels, Belgium, 11–16 July 2021; pp. 1–4.
8. Ciotola, M.; Martinelli, A.; Mazza, A.; Scarpa, G. An Adversarial Training Framework for Sentinel-2 Image Super-Resolution. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 3782–3785. [CrossRef]
9. Peng, D.; Bruzzone, L.; Zhang, Y.; Guan, H.; Ding, H.; Huang, X. SemiCDNet: A Semisupervised Convolutional Neural Network for Change Detection in High Resolution Remote-Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5891–5906. [CrossRef]
10. Gaetano, R.; Amitrano, D.; Masi, G.; Poggi, G.; Ruello, G.; Verdoliva, L.; Scarpa, G. Exploration of multitemporal COSMO-skymed data via interactive tree-structured MRF segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2763–2775. [CrossRef]

11. Vivone, G.; Dalla Mura, M.; Garzelli, A.; Restaino, R.; Scarpa, G.; Ulfarsson, M.O.; Alparone, L.; Chanussot, J. A New Benchmark Based on Recent Advances in Multispectral Pansharpening: Revisiting Pansharpening With Classical and Emerging Pansharpening Methods. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 53–81. [CrossRef]
12. Chavez, P.S.; Kwarteng, A.W. Extracting spectral contrast in Landsat thematic mapper image data using selective principal component analysis. *Photogramm. Eng. Remote Sens.* **1989**, *55*, 339–348.
13. Tu, T.M.; Huang, P.S.; Hung, C.L.; Chang, C.P. A fast intensity hue-saturation fusion technique with spectral adjustment for IKONOS imagery. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 309–312. [CrossRef]
14. Gillespie, A.R.; Kahle, A.B.; Walker, R.E. Color enhancement of highly correlated images. II. Channel ratio and "chromaticity" transformation techniques. *Remote Sens. Environ.* **1987**, *22*, 343–365. [CrossRef]
15. Laben, C.A.; Brower, B.V. Process for Enhancing the Spatial Resolution of Multispectral Imagery Using Pan-Sharpening. U.S. Patent 6011875, 4 January 2000.
16. Ranchin, T.; Wald, L. Fusion of high spatial and spectral resolution images: The ARSIS concept and its implementation. *Photogramm. Eng. Remote Sens.* **2000**, *66*, 49–61.
17. Aiazzi, B.; Alparone, L.; Baronti, S.; Garzelli, A. Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 2300–2312. [CrossRef]
18. Khan, M.M.; Chanussot, J.; Condat, L.; Montanvert, A. Indusion: Fusion of Multispectral and Panchromatic Images Using the Induction Scaling Technique. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 98–102. [CrossRef]
19. Restaino, R.; Mura, M.D.; Vivone, G.; Chanussot, J. Context-Adaptive Pansharpening Based on Image Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 753–766. [CrossRef]
20. Palsson, F.; Sveinsson, J.R.; Ulfarsson, M.O. A New Pansharpening Algorithm Based on Total Variation. *Geosci. Remote Sens. Lett. IEEE* **2014**, *11*, 318–322. [CrossRef]
21. Palsson, F.; Sveinsson, J.R.; Ulfarsson, M.O.; Benediktsson, J.A. Model-Based Fusion of Multi- and Hyperspectral Images Using PCA and Wavelets. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2652–2663. [CrossRef]
22. Vivone, G.; Simões, M.; Dalla Mura, M.; Restaino, R.; Bioucas-Dias, J.M.; Licciardi, G.A.; Chanussot, J. Pansharpening Based on Semiblind Deconvolution. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1997–2010. [CrossRef]
23. Palsson, F.; Ulfarsson, M.O.; Sveinsson, J.R. Model-Based Reduced-Rank Pansharpening. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 656–660. [CrossRef]
24. Masi, G.; Cozzolino, D.; Verdoliva, L.; Scarpa, G. Pansharpening by Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 594. [CrossRef]
25. Yang, J.; Fu, X.; Hu, Y.; Huang, Y.; Ding, X.; Paisley, J. PanNet: A Deep Network Architecture for Pan-Sharpening. In Proceedings of the ICCV, Venice, Italy, 22–29 October 2017. [CrossRef]
26. Masi, G.; Cozzolino, D.; Verdoliva, L.; Scarpa, G. CNN-based Pansharpening of Multi-Resolution Remote-Sensing Images. In Proceedings of the Joint Urban Remote Sensing Event, Dubai, United Arab Emirates, 6–8 March 2017.
27. Yuan, Q.; Wei, Y.; Meng, X.; Shen, H.; Zhang, L. A Multiscale and Multidepth Convolutional Neural Network for Remote Sensing Imagery Pan-Sharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 978–989. [CrossRef]
28. Shao, Z.; Cai, J. Remote Sensing Image Fusion With Deep Convolutional Neural Network. *IEEE J. Sel. Topics Appl. Earth Observ.* **2018**, *11*, 1656–1669. [CrossRef]
29. Vicinanza, M.R.; Restaino, R.; Vivone, G.; Mura, M.D.; Chanussot, J. A Pansharpening Method Based on the Sparse Representation of Injected Details. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 180–184. [CrossRef]
30. Wei, Y.; Yuan, Q.; Shen, H.; Zhang, L. Boosting the Accuracy of Multispectral Image Pansharpening by Learning a Deep Residual Network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1795–1799. [CrossRef]
31. Luo, S.; Zhou, S.; Feng, Y.; Xie, J. Pansharpening via Unsupervised Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4295–4310. [CrossRef]
32. Seo, S.; Choi, J.S.; Lee, J.; Kim, H.H.; Seo, D.; Jeong, J.; Kim, M. UPSNet: Unsupervised Pan-Sharpening Network With Registration Learning Between Panchromatic and Multi-Spectral Images. *IEEE Access* **2020**, *8*, 201199–201217. [CrossRef]
33. Wald, L.; Ranchin, T.; Mangolini, M. Fusion of satellite images of different spatial resolution: Assessing the quality of resulting images. *Photogramm. Eng. Remote Sens.* **1997**, *63*, 691–699.
34. Vitale, S.; Scarpa, G. A Detail-Preserving Cross-Scale Learning Strategy for CNN-Based Pansharpening. *Remote Sens.* **2020**, *12*, 348. [CrossRef]
35. Ciotola, M.; Vitale, S.; Mazza, A.; Poggi, G.; Scarpa, G. Pansharpening by convolutional neural networks in the full resolution framework. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5408717. [CrossRef]
36. Scarpa, G.; Vitale, S.; Cozzolino, D. Target-Adaptive CNN-Based Pansharpening. *IEEE Trans. Geosci. Remote. Sens.* **2018**, *56*, 5443–5457. [CrossRef]
37. Zhang, Y.; Liu, C.; Sun, M.; Ou, Y. Pan-Sharpening Using an Efficient Bidirectional Pyramid Network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5549–5563. [CrossRef]
38. Xu, S.; Zhang, J.; Zhao, Z.; Sun, K.; Liu, J.; Zhang, C. Deep Gradient Projection Networks for Pan-sharpening. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1366–1375. [CrossRef]

39. Deng, L.J.; Vivone, G.; Paoletti, M.E.; Scarpa, G.; He, J.; Zhang, Y.; Chanussot, J.; Plaza, A.J. Machine Learning in Pansharpening: A Benchmark, from Shallow to Deep Networks. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 279–315. [CrossRef]
40. Scarpa, G.; Ciotola, M. Full-Resolution Quality Assessment for Pansharpening. *Remote Sens.* **2022**, *14*, 1808. [CrossRef]
41. Lolli, S.; Alparone, L.; Garzelli, A.; Vivone, G. Haze Correction for Contrast-Based Multispectral Pansharpening. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2255–2259. [CrossRef]
42. Garzelli, A.; Nencini, F.; Capobianco, L. Optimal MMSE pan sharpening of very high resolution multispectral images. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 228–236. [CrossRef]
43. Garzelli, A. Pansharpening of Multispectral Images Based on Nonlocal Parameter Optimization. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2096–2107. [CrossRef]
44. Vivone, G. Robust Band-Dependent Spatial-Detail Approaches for Panchromatic Sharpening. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6421–6433. [CrossRef]
45. Aiazzi, B.; Baronti, S.; Selva, M. Improving component substitution pansharpening through multivariate regression of MS+Pan data. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3230–3239. [CrossRef]
46. Choi, J.; Yu, K.; Kim, Y. A New Adaptive Component-Substitution-Based Satellite Image Fusion by Using Partial Replacement. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 295–309. [CrossRef]
47. Otazu, X.; Gonzalez-Audicana, M.; Fors, O.; Nunez, J. Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 2376–2385. [CrossRef]
48. Alparone, L.; Garzelli, A.; Vivone, G. Intersensor Statistical Matching for Pansharpening: Theoretical Issues and Practical Solutions. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4682–4695. [CrossRef]
49. Vivone, G.; Restaino, R.; Chanussot, J. Full Scale Regression-Based Injection Coefficients for Panchromatic Sharpening. *IEEE Trans. Image Process.* **2018**, *27*, 3418–3431. [CrossRef] [PubMed]
50. Vivone, G.; Restaino, R.; Chanussot, J. A Regression-Based High-Pass Modulation Pansharpening Approach. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 984–996. [CrossRef]
51. Alparone, L.; Wald, L.; Chanussot, J.; Thomas, C.; Gamba, P.; Bruce, L. Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S Data-Fusion Contest. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3012–3021. [CrossRef]
52. Restaino, R.; Vivone, G.; Dalla Mura, M.; Chanussot, J. Fusion of Multispectral and Panchromatic Images Based on Morphological Operators. *IEEE Trans. Image Process.* **2016**, *25*, 2882–2895. [CrossRef]
53. He, L.; Rao, Y.; Li, J.; Chanussot, J.; Plaza, A.; Zhu, J.; Li, B. Pansharpening via detail injection based convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1188–1204. [CrossRef]
54. Deng, L.J.; Vivone, G.; Jin, C.; Chanussot, J. Detail injection-based deep convolutional neural networks for pansharpening. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 6995–7010. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.