Supplementary Material of "Laboring on less labors: RPCA Paradigm for Pan-sharpening"

Honghui Xu[1,2], Chuangjie Fang[1], Yibin Wang[3], Jie Wu[1], Jianwei Zheng[1*]

[1]Zhejiang University of Technology [2]Taizhou University [3]Fudan University

xhh9609@gmail.com, fangcj@zjut.edu.cn, yibinwang1121@163.com, {wuj,zjw}@zjut.edu.cn

In this supplementary material, we primarily provide related work on pansharpening, more details for solving the proposed RUN modell, and real-world experiments.

1. Related Work

In this section, we introduce most related methods for pansharpening, generally following two main branches, including DL-based and DUN-based methods.

1.1. DL-based Pansharpening

Due to the great achievements in various computer vision communities, CNNs have naturally emerged as a mainstream technique for pansharpening, with powerful modeling capabilities in learning mapping relationships among PAN, LRMS, and HRMS. Representatively, He et al. [7] crafted a convolution-based module, specifically aiming at the recovery of spatial details that were absent in LRMS images. By leveraging the strengths of both LRMS and PAN images, Ref. [23] employed two distinct CNNs that were tailored to extract complementary features. Hou et al. [8] proposed a new source-discriminative adaption convolution and bidomain modeling for pansharpening, broadcasting a general framework for high resolution learning.

With the initial success in language processing, Transformer has now demonstrated exceptional proficiency in capturing long-range dependencies within the context of image restoration. Following this route, Lu et al. [16] proposed a model with lightweight CNNs and Transformers combined to address the image super-resolution task. Bandara et al. [3] engineered a novel fusion network of textural-spectral features for pansharpening, which transfers high-resolution textural features from PANs to LRMS spectral features through a multi-head soft-attention mechanism. Lately, with the aid of a dual Transformer architecture, Quan et al. [17] proposed an effective semantic restoration network for RS pansharpening. Zhang et al. [22] introduced a novel high-frequency wavelet network that capitalizes on the spatial–frequency interaction and frequency division capabilities inherent in wavelet transform.

Yet with exceptional results delivered, the purely network solutions suffer notoriously from the black-box nature, whose performance depends heavily on massive attempts of error avoidance even given large-scale datasets, leaving along the poor interpretability.

1.2. DUN-based Pansharpening

To integrate strengths while mitigating the limitations of model-based and DL-based approaches, DUNs primarily tailor DL modules to the investigated pansharpening challenge, guided by the theoretical philosophy. For instance, Xu et al. [20] developed GPPNN, a novel network derived from the resolution of dual optimization problems, which is then efficiently optimized through a gradient projection. A unique framework named PanCSC-Net, as introduced by [4], employed convolutional sparse coding for better distinction between shared and distinct characteristics of LRMS and PAN. Later, Yang et al. [21] introduced a highly interpretable deep neural network called memory-augmented conditional unfolding. Li et al. [12] developed a Transformer variant for DUN and customized a local-global denoiser to efficiently model complementary dependencies. Lately, another DUN with high interpretability was proposed by Li et al. [13], which is based solely on the learning of convolutional dictionaries in the encoding stage. Unfortunately, all current solutions require a significant amount of effort designing simulations for spatial and spectral response matrices, with expected quality rarely guaranteed. In addition, simulations are often dominated by black-box networks and suffer from poor interpretability.

2. ADMM-based Unfolding Optimization

In this section, we simply employ the typical ADMM scheme, yet put more efforts on network architecture during unfolding stages. Based on the unconstrained form of

*Corresponding author.

Eq. (5), we can obtain:

$$L_{\mu_i}(\boldsymbol{S},\boldsymbol{X},\boldsymbol{T},\boldsymbol{M}_1,\boldsymbol{M}_2)=\alpha\|\boldsymbol{S}\|_1+\lambda J_1(\boldsymbol{T})+\frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{R}\boldsymbol{X}\|_F^2 + \frac{\mu_1}{2}\|\bar{\boldsymbol{Y}}-(\boldsymbol{X}+\boldsymbol{S})+\frac{\boldsymbol{M}_1}{\mu_1}\|_F^2+\frac{\mu_2}{2}\|\boldsymbol{T}-\boldsymbol{X}+\frac{\boldsymbol{M}_2}{\mu_2}\|_F^2 \tag{1}$$

By alternately updating one variable with the others fixed, problem (1) can be solved separately.

Algorithm 1 ADMM for RPCA-based Pansharpening

1: **Input:** LRMS $\boldsymbol{Y}$; PAN $\boldsymbol{Z}$; Parameter α; Maximum iteration T.
2: **Output:** HRMS $\boldsymbol{X}$.
3: **Initialization:** Interpolated LRMS $\bar{\boldsymbol{Y}}$, set $t=0$.
4: **while** $t \leqslant T$ **do**
5: Update $\boldsymbol{S}^{t+1}$ via

$$\boldsymbol{S}^{t+1}=\text{shrink}\left(\bar{\boldsymbol{Y}}-\boldsymbol{X}^t+\frac{\boldsymbol{M}_1^t}{\mu_1},\frac{\alpha}{\mu_1}\right)$$

6: Update $\boldsymbol{T}^{t+1}$ via

$$\boldsymbol{T}^{t+1}=\textbf{proxNet}\left(\boldsymbol{X}^t-\frac{\boldsymbol{M}_2^t}{\mu_2}\right)$$

7: Update $\boldsymbol{X}^{t+1}$ via

$$\begin{aligned}\boldsymbol{X}^{t+1}=\boldsymbol{X}^t&-\gamma\boldsymbol{R}^T\left(\boldsymbol{Z}-\boldsymbol{R}\boldsymbol{X}^t\right)\\&-\gamma\mu_2\left(\boldsymbol{X}^t-\boldsymbol{T}^{t+1}-\frac{\boldsymbol{M}_2^t}{\mu_2}\right)\\&-\gamma\mu_1\left(\boldsymbol{X}^t-\bar{\boldsymbol{Y}}+\boldsymbol{S}^{t+1}-\frac{\boldsymbol{M}_1^t}{\mu_1}\right)\end{aligned}$$

8: Update $\boldsymbol{M}_1^{t+1}$ and $\boldsymbol{M}_2^{t+1}$ via

$$\begin{aligned}\boldsymbol{M}_1^{t+1}&=\boldsymbol{M}_1^t+\mu_1\left(\bar{\boldsymbol{Y}}-\left(\boldsymbol{X}^{t+1}+\boldsymbol{S}^{t+1}\right)\right)\\\boldsymbol{M}_2^{t+1}&=\boldsymbol{M}_2^t+\mu_2\left(\boldsymbol{T}^{t+1}-\boldsymbol{X}^{t+1}\right)\end{aligned}$$

9: $t\leftarrow t+1$
10: **end while**

Figure 1. Illustration of spectral degradation matrices $\boldsymbol{R}$ and $\boldsymbol{R}^T$.

Step 1: For $\boldsymbol{S}$-subproblem. By extracting all terms containing $\boldsymbol{S}$ from problem (1), we need to solve:

$$\boldsymbol{S}^{t+1}=\arg\min_{\boldsymbol{S}}\alpha\|\boldsymbol{S}\|_1+\frac{\mu_1}{2}\|\bar{\boldsymbol{Y}}-(\boldsymbol{X}^t+\boldsymbol{S})+\frac{\boldsymbol{M}_1^t}{\mu_1}\|_F^2, \tag{2}$$

which has the following solution:

$$\boldsymbol{S}^{t+1}=\text{shrink}\left(\bar{\boldsymbol{Y}}-\boldsymbol{X}^{t+1}+\frac{\boldsymbol{M}_1^t}{\mu_1},\frac{\alpha}{\mu_1}\right) \tag{3}$$

in which shrink is the thresholding operator.

Step 2: For $\boldsymbol{T}$-subproblem. By removing the terms that are irrelevant to $\boldsymbol{T}$, we can deduce

$$\boldsymbol{T}^{t+1}=\lambda J_1(\boldsymbol{T})+\frac{\mu_2}{2}\|\boldsymbol{T}-\boldsymbol{X}^t+\frac{\boldsymbol{M}_2^t}{\mu_2}\|_F^2 \tag{4}$$

Eq. (4) is a deterministic approximate operator given a known prior J_1. Unfortunately, the uncertainty of the J_1 function also causes the inaccessibility of any closed-form solutions. Accordingly, we propose a low-rank network (LRNet) as the prior extractor for joint implicit-explicit usage, which can typically be expressed as:

$$\boldsymbol{T}^{t+1}=\textbf{proxNet}(\boldsymbol{X}^t-\frac{\boldsymbol{M}_2^t}{\mu_2}) \tag{5}$$

where **proxNet** is an approximate operator replaced by the proposed network architecture.

Step 3: For $\boldsymbol{X}$-subproblem. The $\boldsymbol{X}$-related subproblem at the t-th iteration is

$$\begin{aligned}&\boldsymbol{X}^{t+1}=\tfrac{1}{2}\|\boldsymbol{Z}-\boldsymbol{R}\boldsymbol{X}\|_F^2+\tfrac{\mu_1}{2}\|\bar{\boldsymbol{Y}}-(\boldsymbol{X}+\boldsymbol{S}^{t+1})+\tfrac{\boldsymbol{M}_1^t}{\mu_1}\|_F^2\\&+\tfrac{\mu_2}{2}\|\boldsymbol{T}^{t+1}-\boldsymbol{X}+\tfrac{\boldsymbol{M}_2^t}{\mu_2}\|_F^2\end{aligned} \tag{6}$$

Obviously, Eq. (6) is already differentiable, the gradient descent scheme can be borrowed as a solver.

$$\begin{aligned}&\boldsymbol{X}^{t+1}=\boldsymbol{X}^t-\gamma\boldsymbol{R}^T\left(\boldsymbol{Z}-\boldsymbol{R}\boldsymbol{X}^t\right)-\gamma\mu_2\left(\boldsymbol{X}^t-\boldsymbol{T}^{t+1}-\tfrac{\boldsymbol{M}_2^t}{\mu_2}\right)\\&-\gamma\mu_1\left(\boldsymbol{X}^t-\bar{\boldsymbol{Y}}+\boldsymbol{S}^{t+1}-\tfrac{\boldsymbol{M}_1^t}{\mu_1}\right)\end{aligned} \tag{7}$$

where $\gamma > 0$ is the stage-specific learnable step size. What's more, a convolution is employed in place of the spectral degradation matrix $\boldsymbol{R}$ to reduce channels from S to 1, and another point Conv is utilized as the matrix $\boldsymbol{R}^T$ for the corresponding inverse channel increase, as depicted in Fig. 1.

Step 4: Update of the Lagrangian parameters. Under the principle of general ADMM, we can update the Lagrangian multipliers in parallel as

$$\begin{cases}\boldsymbol{M}_1^{t+1}=\boldsymbol{M}_1^t+\mu_1(\bar{\boldsymbol{Y}}-(\boldsymbol{X}^{t+1}+\boldsymbol{S}^{t+1}))\\\boldsymbol{M}_2^{t+1}=\boldsymbol{M}_2^t+\mu_2(\boldsymbol{T}^{t+1}-\boldsymbol{X}^{t+1})\end{cases} \tag{8}$$

Summarizing the aforementioned descriptions, we present the entire ADMM-based solution process for solving the problem (1) in Algorithm 1.

3. Proof of Theorem 1

We aim to show that the combination of depthwise convolution and SoftShrink effectively approximates Singular Value Thresholding (SVT). This proof follows by analyzing the spectral transformation induced by depthwise convolution and demonstrating how SoftShrink enforces singular value shrinkage.

3.1. Step 1: Depthwise Convolution as a Localized Spectral Transformation

Consider a given input tensor $\mathcal{T} \in \mathbb{R}^{H\times W\times C}$. Depthwise convolution operates **independently** on each channel, applying a spatial filter to each slice $\mathcal{T}^{(k)} \in \mathbb{R}^{H\times W}$ for $k = 1, ..., C$. Formally, depthwise convolution is defined as:

$$\mathcal{F}_\theta(\mathcal{T}) = \mathcal{T} \times_1 \boldsymbol{K}_h \times_2 \boldsymbol{K}_w, \tag{9}$$

where $\boldsymbol{K}_h, \boldsymbol{K}_w \in \mathbb{R}^{3\times 3}$ are learnable convolutional kernels applied along the height and width dimensions, respectively. Expanding for an individual channel $\mathcal{T}^{(k)}$, the operation can be rewritten in matrix form as:

$$\mathcal{T}^{(k)}_{\text{conv}} = \boldsymbol{K}_h \mathcal{T}^{(k)} \boldsymbol{K}_w^\top. \tag{10}$$

This formulation resembles a bilinear transformation of $\mathcal{T}^{(k)}$ using the filters $\boldsymbol{K}_h$ and $\boldsymbol{K}_w$, which can be interpreted as a localized spectral transformation. Under the assumption that these filters approximate orthonormal decompositions, this convolutional operation can be rewritten in a singular value form:

$$\mathcal{F}_\theta(\mathcal{T}) = \mathcal{U}\boldsymbol{\Sigma}\mathcal{V}^\top. \tag{11}$$

Thus, the depthwise convolution extracts dominant spatial structures, approximating a low-rank projection.

3.2. Step 2: SoftShrink as Singular Value Thresholding

The SoftShrink function is defined as a nonlinear element-wise thresholding function:

$$\mathcal{H}_\tau(x) = \begin{cases} x - \tau, & x > \tau, \\ 0, & -\tau \le x \le \tau, \\ x + \tau, & x < -\tau. \end{cases} \tag{12}$$

When applied to the singular values of $\boldsymbol{\Sigma}$, the function modifies each singular value σ_i as:

$$\tilde{\sigma}_i = \max(\sigma_i - \tau, 0). \tag{13}$$

This operation is equivalent to the Singular Value Thresholding (SVT) step in low-rank approximation techniques, where only singular values greater than τ are retained. Applying SoftShrink to the singular values in the decomposition:

$$\tilde{\boldsymbol{\Sigma}} = \text{SoftShrink}(\boldsymbol{\Sigma}, \tau), \tag{14}$$

where each singular value σ_i is thresholded.

3.3. Step 3: Rank Preservation and Reduction

From Step 1, depthwise convolution approximates a localized singular value decomposition:

$$\mathcal{F}_\theta(\mathcal{T}) = \mathcal{U}\boldsymbol{\Sigma}\mathcal{V}^\top. \tag{15}$$

Applying SoftShrink to the singular values leads to:

$$\mathcal{T}^\star = \mathcal{U}\tilde{\boldsymbol{\Sigma}}\mathcal{V}^\top, \quad \tilde{\sigma}_i = \max(\sigma_i - \tau, 0). \tag{16}$$

Since thresholding eliminates small singular values, we obtain the rank inequality:

$$\text{rank}(\tilde{\boldsymbol{\Sigma}}) \le \text{rank}(\boldsymbol{\Sigma}). \tag{17}$$

Thus, the rank of $\mathcal{T}^\star$ satisfies:

$$\text{rank}(\mathcal{T}^\star) \le \text{rank}(\mathcal{T}). \tag{18}$$

This completes the proof.

4. Real-World Experiments

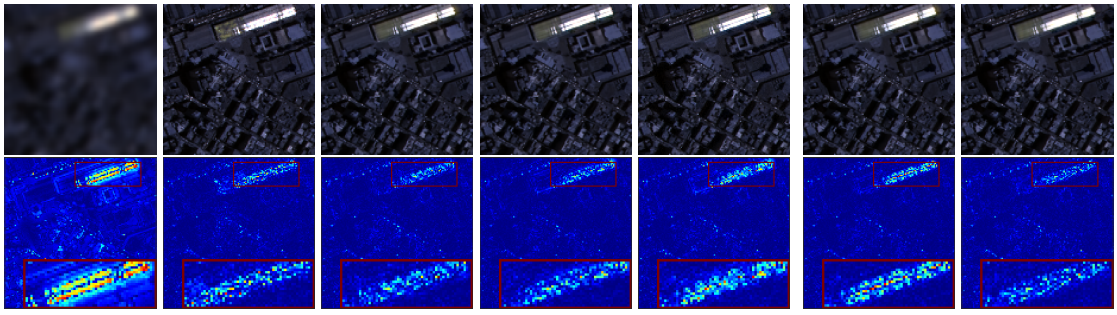
4.1. Quality Metrics and Competitors

To comprehensively assess the quantitative results, six widely used metrics, *i.e.*, Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), Spectral Angle Mapper (SAM), Erreur Relative Globale Admensionnelle de Synthese (ERGSA), as well as the universal image quality indices for 4-band (Q4) and 8-band images (Q8) [6], are adopted.

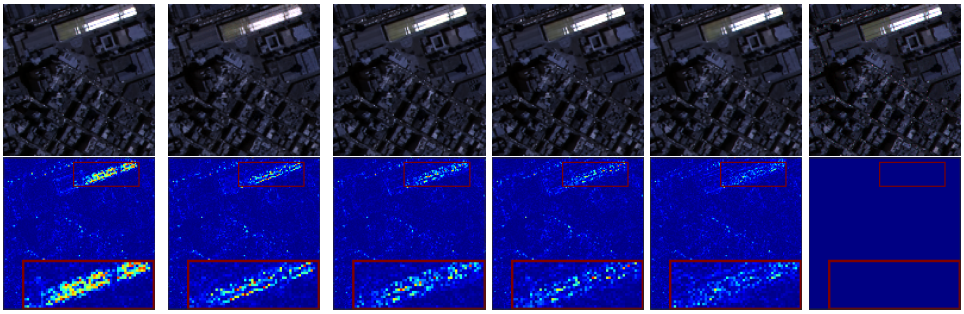
Overall, thirteen cutting-edge approaches are selected as competitors, including three model-based methods, *i.e.*, GSA [1], SFIM [14], as well as Wavelet [11], and ten DL-based solutions, *i.e.*, SFIIN [26], MutInf [27], MDCUN [21], LGTEUN [12], BiMPan [8], SSDBPN [23], PAPS [9], WINet[22], DISPNet [19], and MSAN [15]. The practical settings of all compared methods follow the source codes provided by the original authors. The experiments are carried out in the same environment, *i.e.*, PyTorch framework with a single NVIDIA GTX 3090 GPU. The specific settings of our proposal are as follows: l_1 loss, Adam optimizer (0.9, 0.999), Epoch (500 for GF2, WV2 and 140 for WV3), Batch size (4), Initial learning rate (1.5×10^{-3}), decay rate (0.85 per 100 epoch). The recovered example from WV-3 and the corresponding error maps are shown in Fig. 2.

Table 1. Average results at full resolution. The best and second best values are respectively highlighted by red and blue colors.

Method	WorldView-3			GaoFen-2		
	$D_\lambda\downarrow$	$D_s\downarrow$	QNR$\uparrow$	$D_\lambda\downarrow$	$D_s\downarrow$	QNR$\uparrow$
SFIIN$_{22}$	0.0198	0.0352	0.9458	0.0687	0.1876	0.7557
MutInf$_{22}$	0.0163	0.0420	0.9423	0.0755	0.1762	0.7613
MDCUN$_{22}$	0.0747	0.1673	0.7708	0.0712	0.1938	0.7712
LGTEUN$_{23}$	0.0162	0.0310	0.9532	0.0696	0.1981	0.7457
BiMPan$_{23}$	0.0298	0.0305	0.9463	0.0644	0.1419	0.8120
SSDBPN$_{23}$	0.0262	0.0581	0.9181	0.0663	0.1575	0.7861
PAPS$_{24}$	0.0365	0.0405	0.9244	0.0682	0.2144	0.7317
WINet$_{24}$	0.0214	0.0302	0.9451	0.0682	0.1406	0.7936
DISPNet$_{24}$	0.0136	0.0387	0.9501	0.0687	0.1465	0.7974
MSAN$_{25}$	0.0187	0.0279	0.9539	0.0661	0.1651	0.8058
RUN	0.0254	0.0274	0.9535	0.0660	0.1386	0.8131



(a) LRMS (b) SFIIN[26] (c) MutInf[27] (d) MDCUN[21] (e) LGTEUN[12] (f) BiMPan[8] (g) SSDBPN[23]

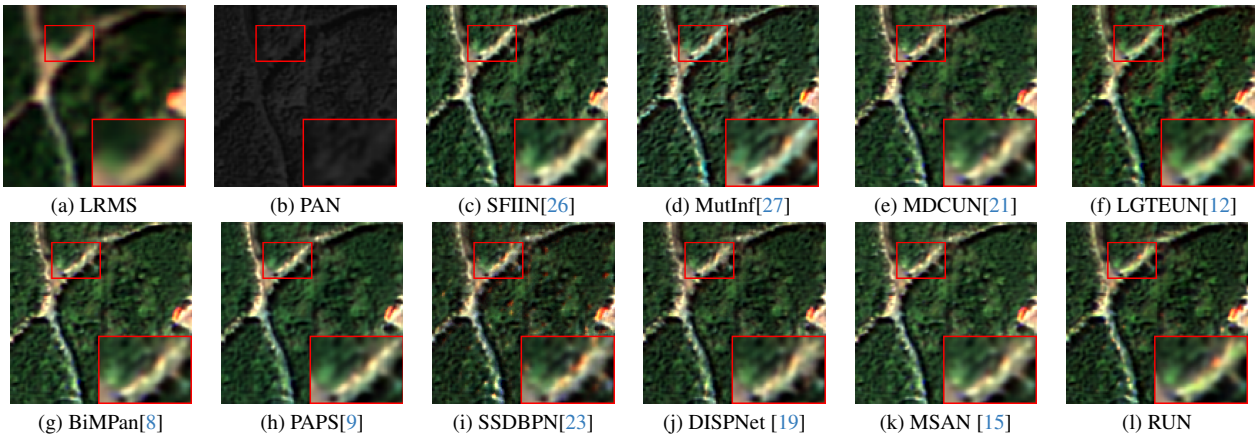


(h) PAPS[9] (i) WINet[22] (j) DISPNet[19] (k) MSAN[15] (l) RUN (m) Ground truth

Figure 2. Visual comparisons on typical WV-3 data.

4.2. Results on Full-Scale Datasets

In this section, we present the outcomes of our method on a full-scale dataset and the comparative analysis against alternative approaches. Given the absence of reference images, we directly feed the full-scale images into the model, which is previously trained with the reduced-scale datasets, to obtain pansharpening outputs. We investigated the fusion performance of all algorithms at full resolution. In this scenario, we employed hybrid quality without reference (QNR) [2], Spatial Distortion Index D_s [18], and Spectral Distortion Index D_λ [10], as criteria. In each dataset, we test with PAN images of 512×512 and LRMS images of 128×128. From Table 1, one can easily see that our proposed RUN performs on par with other algorithms on the WV-3 dataset, yet achieves the best results on the GF-2 dataset. Overall, this indicates that it possesses excellent spectral and spatial preservation capabilities. The visual comparison against reference methods at full resolution is presented in Fig. 3, where we show a representative GF-2 case. Among these deep learning-based methods, LightNet exhibits relatively poor spectral information retention capabilities. Although SFIIN, MDCUN, and SSDBPN show slightly better perfor-



(a) LRMS (b) PAN (c) SFIIN[26] (d) MutInf[27] (e) MDCUN[21] (f) LGTEUN[12]

(g) BiMPan[8] (h) PAPS[9] (i) SSDBPN[23] (j) DISPNet [19] (k) MSAN [15] (l) RUN

Figure 3. Visual comparisons on typical GF-2 data.

Table 2. Ablation study with PSNR and parameters (M).

Ablation Setting	PSNR↑	Params (M)↓
w/o SOC	30.0245	1.3412
DWT+DWConv	32.3071	1.3412
Nonlinear+SVT	N/A	3.4408
Dimension L-3×	32.6097	1.5030
Dimension L-4×	32.6204	1.6648
Dimension L-5×	32.6241	1.8265
Block number-1	32.3974	0.9895
Block number-3	32.3870	1.6929
Only Conv 3×3	32.1025	1.3611
Full Model (Ours)	32.6064	1.3412

mance, their results exhibit blurred rooftop edges in road areas. Compared to other methods, our approach achieves optimal fusion results even without GT guidance, highlighting its superior performance.

5. Parameter analysis and ablation experiments

In Table 2, we conduct two key experiments: (1) a comparison between DWT transformation and DWConv, and (2) an analysis of nonlinear transformation combined with traditional SVT. The results indicate that DWT, as a linear transformation, exhibits lower feature extraction efficiency compared to nonlinear transformation, while traditional SVT incurs a significant parameter overhead due to the necessity of performing SVD at each layer.

For the parameter L in Eq. (7) of the main text, we further explore its setting within the range of 3-5×. Our findings suggest that increasing L enhances the richness of information representation but also leads to a substantial increase in the number of parameters. Empirical results demonstrate that setting $L = 2$ provides a favorable balance between performance and computational efficiency, further validating the effectiveness of combining nonlinear transformation with SVT. Additionally, when replacing the multi-scale CNN with a standard 3×3 convolution, we observe a notable performance degradation, highlighting the critical role of multi-scale feature aggregation in achieving optimal results.

References

[1] Bruno Aiazzi, Stefano Baronti, and Massimo Selva. Improving component substitution pansharpening through multivariate regression of ms + pan data. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3230–3239, 2007. 3

[2] B Aiazzi, Luciano Alparone, S Baronti, R Carlà, Andrea Garzelli, and L Santurri. Full-scale assessment of pansharpening methods and data products. In *Image and Signal Processing for Remote Sensing XX*, page 924402. SPIE, 2014. 4

[3] Wele Gedara Chaminda Bandara and Vishal M Patel. Hypertransformer: A textural and spectral feature fusion transformer for pansharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1767–1777, 2022. 1

[4] Xiangyong Cao, Xueyang Fu, Danfeng Hong, Zongben Xu, and Deyu Meng. Pancsc-net: A model-driven deep unfolding method for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. 1

[5] Zhi-Xuan Chen, Cheng Jin, Tian-Jing Zhang, Xiao Wu, and Liang-Jian Deng. Spanconv: A new convolution via spanning kernel space for lightweight pansharpening. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–7, 2022.

[6] Andrea Garzelli and Filippo Nencini. Hypercomplex quality assessment of multi/hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 6(4):662–665, 2009. 3

[7] Lin He, Yizhou Rao, Jun Li, Jocelyn Chanussot, Antonio Plaza, Jiawei Zhu, and Bo Li. Pansharpening via detail injection based convolutional neural networks. *IEEE Journal of*

Selected Topics in Applied Earth Observations and Remote Sensing, 12(4):1188–1204, 2019. 1

[8] Junming Hou, Qi Cao, Ran Ran, Che Liu, Junling Li, and Liang-jian Deng. Bidomain modeling paradigm for pansharpening. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, pages 347–357, 2023. 1, 3, 4, 5

[9] Yanan Jia, Qiming Hu, Renwei Dian, Jiayi Ma, and Xiaojie Guo. Paps: Progressive attention-based pan-sharpening. *IEEE/CAA Journal of Automatica Sinica*, 11(2):391–404, 2024. 3, 4, 5

[10] Muhammad Murtaza Khan, Luciano Alparone, and Jocelyn Chanussot. Pansharpening quality assessment using the modulation transfer functions of instruments. *IEEE Transactions on Geoscience and Remote Sensing*, 47(11):3880–3891, 2009. 4

[11] Roger L King and Jianwen Wang. A wavelet based algorithm for pan sharpening landsat 7 imagery. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 849–851. IEEE, 2001. 3

[12] Mingsong Li, Yikun Liu, Tao Xiao, Yuwen Huang, and Gongping Yang. Local-global transformer enhanced unfolding network for pan-sharpening. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1071–1079, 2023. 1, 3, 4, 5

[13] Zixu Li, Genji Yuan, and Jinjiang Li. Ducd: Deep unfolding convolutional-dictionary network for pansharpening remote sensing image. *Expert Systems with Applications*, 249: 123589, 2024. 1

[14] JG Liu. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18): 3461–3472, 2000. 3

[15] Hangyuan Lu, Yong Yang, Shuying Huang, Rixian Liu, and Huimin Guo. Msan: Multiscale self-attention network for pansharpening. *Pattern Recognition*, 162:111441, 2025. 3, 4, 5

[16] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 457–466, 2022. 1

[17] Wuzhou Quan and Wen Guo. Dtempan: Dual texture-edge maintaining transformer for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023. 1

[18] Giuseppe Scarpa and Matteo Ciotola. Full-resolution quality assessment for pansharpening. *Remote Sensing*, 14(8):1808, 2022. 4

[19] Hebaixu Wang, Meiqi Gong, Xiaoguang Mei, Hao Zhang, and Jiayi Ma. Deep unfolded network with intrinsic supervision for pan-sharpening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5419–5426, 2024. 3, 4, 5

[20] Shuang Xu, Jiangshe Zhang, Zixiang Zhao, Kai Sun, Junmin Liu, and Chunxia Zhang. Deep gradient projection networks for pan-sharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1366–1375, 2021. 1

[21] Gang Yang, Man Zhou, Keyu Yan, Aiping Liu, Xueyang Fu, and Fan Wang. Memory-augmented deep conditional unfolding network for pan-sharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1788–1797, 2022. 1, 3, 4, 5

[22] Jie Zhang, Xuanhua He, Keyu Yan, Ke Cao, Rui Li, Chengjun Xie, Man Zhou, and Danfeng Hong. Pansharpening with wavelet-enhanced high-frequency information. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024. 1, 3, 4

[23] Kai Zhang, Anfei Wang, Feng Zhang, Wenbo Wan, Jiande Sun, and Lorenzo Bruzzone. Spatial-spectral dual backprojection network for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023. 1, 3, 4, 5

[24] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. Panformer: A transformer based model for pan-sharpening. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[25] Man Zhou, Jie Huang, Yanchi Fang, Xueyang Fu, and Aiping Liu. Pan-sharpening with customized transformer and invertible neural network. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3553–3561, 2022.

[26] Man Zhou, Jie Huang, Keyu Yan, Hu Yu, Xueyang Fu, Aiping Liu, Xian Wei, and Feng Zhao. Spatial-frequency domain information integration for pan-sharpening. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 274–291. Springer, 2022. 3, 4, 5

[27] Man Zhou, Keyu Yan, Jie Huang, Zihe Yang, Xueyang Fu, and Feng Zhao. Mutual information-driven pan-sharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, 2022. 3, 4, 5