A General Adaptive Dual-level Weighting Mechanism for Remote Sensing Pansharpening

Jie Huang[1*] Haorui Chen[1*] Jiaxuan Ren[1] Siran Peng[2, 3] Liangjian Deng[1†]
[1]University of Electronic Science and Technology of China
[2]Institute of Automation, Chinese Academy of Sciences
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences
{jayhuang,hrchen,jiaxuan.ren}@std.uestc.edu.cn, liangjian.deng@uestc.edu.cn

Abstract

Currently, deep learning-based methods for remote sensing pansharpening have advanced rapidly. However, many existing methods struggle to fully leverage feature heterogeneity and redundancy, thereby limiting their effectiveness. We use the covariance matrix to model the feature heterogeneity and redundancy and propose Correlation-Aware Covariance Weighting (CACW) to adjust them. CACW captures these correlations through the covariance matrix, which is then processed by a nonlinear function to generate weights for adjustment. Building upon CACW, we introduce a general adaptive dual-level weighting mechanism (ADWM) to address these challenges from two key perspectives, enhancing a wide range of existing deep-learning methods. First, Intra-Feature Weighting (IFW) evaluates correlations among channels within each feature to reduce redundancy and enhance unique information. Second, Cross-Feature Weighting (CFW) adjusts contributions across layers based on inter-layer correlations, refining the final output. Extensive experiments demonstrate the superior performance of ADWM compared to recent state-of-the-art (SOTA) methods. Furthermore, we validate the effectiveness of our approach through generality experiments, redundancy visualization, comparison experiments, key variables and complexity analysis, and ablation studies. Our code is available at https://github.com/Jie-1203/ADWM.

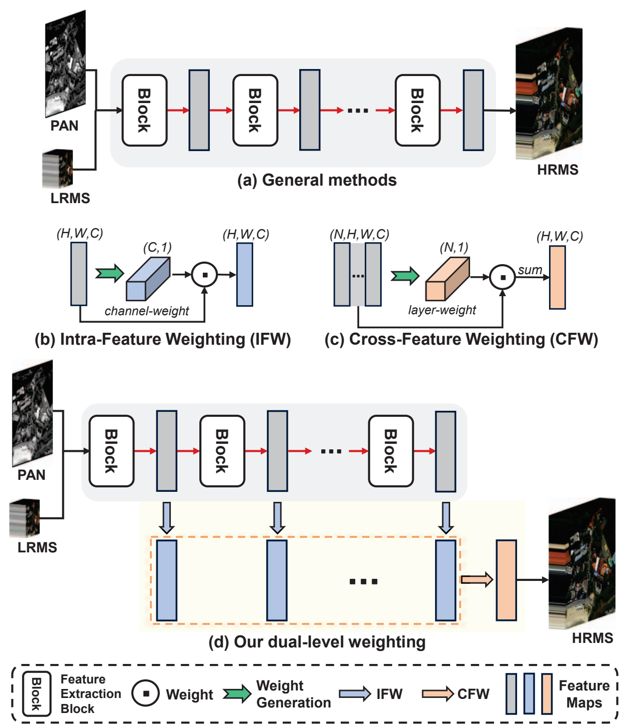

Figure 1. Application of our dual-level weighting mechanism within the existing methods. (a) General methods. (b) Intra-Feature Weighting (IFW): weighting different channels within a single feature. (c) Cross-Feature Weighting (CFW): weighting features at different depths to obtain the final result. (d) Our dual-level weighting combines both IFW and CFW to fully unlock the potential of the original networks.

1. Introduction

The application of high-resolution multispectral (HRMS) images is expanding rapidly, with uses in fields such as object detection [8, 28], change detection [3, 23], unmixing [4], and classification [6, 7]. However, due to technological limitations, satellites are typically only able to capture low-resolution multispectral (LRMS) images alongside high-resolution panchromatic (PAN) images. In this setup, PAN images provide high spatial resolution, whereas LRMS images offer rich spectral information. To generate HRMS images with both high spatial and spectral resolutions, the pansharpening technique was introduced and has seen continuous development.

*Equal contribution.
†Corresponding author.

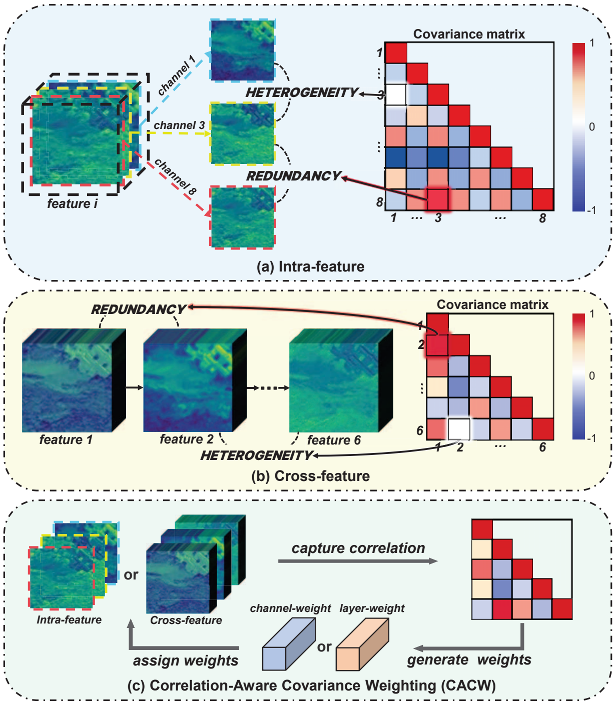

Figure 2. Feature heterogeneity and redundancy correspond to the covariance matrix: darker colors indicate stronger correlations and redundancy, while lighter colors suggest weaker correlations and more heterogeneity. (a) Intra-feature: different channels within a feature. (b) Cross-feature: features at different depths. (c) CACW leverages intra- and cross-feature correlations to generate weights and adjust features accordingly.

Over recent decades, pansharpening methods have evolved considerably, transitioning from traditional approaches to modern deep learning-based techniques. Traditional methods encompass component substitution (CS) [9, 33], multi-resolution analysis (MRA) [34, 35], and variational optimization (VO) [13, 32]. With advancements in hardware and software, deep learning-based methods [12, 14, 41, 45] have shown significant promise in addressing pansharpening challenges. Recent studies focusing on fusion tasks [22, 26, 39] highlight the benefits of sequential feature extraction to achieve effective data fusion. This keeps the continuity with the tradition of using sequential feature extraction as shown in Fig. 1 (a) for achieving improved fusion in pansharpening.

However, the above methods often overlook a crucial characteristic across networks: feature heterogeneity and redundancy, which occur in two dimensions as shown in Fig. 2. In the intra-feature dimension, PAN and LRMS images contain redundant and heterogeneous information, with multiple bands in LRMS images exhibiting similar redundancy [16]. In the cross-feature dimension, shallow features capture low-level details like edges and textures, while deeper features become abstract, encoding semantic information [42]. These features vary by depth but also exhibit redundancy. Overall, previous methods fail to comprehensively address these issues, limiting refined fusion.

To overcome the limitations of previous methods, we propose an Adaptive Dual-level Weighting Mechanism (ADWM). Our method can be seamlessly plugged into the network involving sequential feature processing to fully unleash the potential of the original network, see Fig. 1. Specifically, our first level of weighting, termed Intra-Feature Weighting (IFW), focuses on optimizing the internal structure of intermediate features through adaptive weighting. This step emphasizes the importance of each channel individually, allowing for a refined adjustment that accounts for the feature heterogeneity and redundancy channels themselves. Following this, the second level of weighting, Cross-Feature Weighting (CFW), dynamically adjusts the contribution of features from different layers based on inter-layer correlations. This approach allows features at various depths to contribute adaptively to the final output, ensuring a balanced integration of deep-shallow features across the network. Unlike DenseNet [21], which directly concatenates feature maps from different layers, CFW adjusts each layer's influence based on its relevance to the final output. CFW balances shallow and deep layer contributions, enhancing representation precision and fidelity.

To achieve the above process, we need a weighting method that can finely capture relationships between features. SENet [20] uses global pooling for channel-level weights, applying high-level adaptive weighting but oversimplifies feature compression, which leads to a loss of inter-channel relationships. SRANet [27] employs a self-residual attention mechanism for dynamic channel weights but lacks effective redundancy compression, leading to suboptimal resource use. *Different from previous benchmark attention-based methods, such as channel attention [27, 38], spatial attention [20], etc., our method leverages the correlations within the covariance matrix, effectively capturing feature heterogeneity and redundancy, forming the proposed Correlation-Aware Covariance Weighting (CACW).* The use of the covariance matrix is inspired by Principal Component Analysis (PCA) [1], which utilizes the covariance matrix to capture information correlation for dimensionality reduction. Specifically, as shown in Fig. 2 (c), we first compute a covariance matrix to capture inter- or cross-feature correlations, reflecting feature heterogeneity and redundancy. Subsequently, we further process this covariance matrix through a nonlinear transformation enabling the generation of weights for feature adjustment. By leveraging these weights, the model identifies the importance of different channels or layers, focusing on essential components while effectively reducing redundancy.

To sum up, the contributions of this work are as follows:

1. We use covariance to measure feature heterogeneity and redundancy and propose CACW, which utilizes the corre-

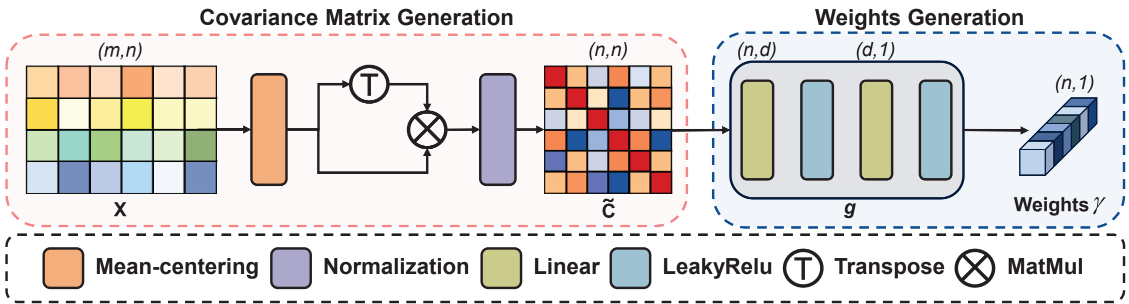

Figure 3. The CACW structure is illustrated. First, we compute the covariance matrix $\tilde{C}$ based on the correlations among the n columns of X. Then, this covariance matrix is passed through a nonlinear function g, which generates the resulting weights.

lations within the covariance matrix to adaptively generate weights. This weighting strategy can also be extended to applications beyond pansharpening.

2. We propose an adaptive dual-level weighting mechanism (ADWM). This mechanism uses IFW to adjust the importance of different channels in the intra-feature dimension and applies CFW to adaptively modulate the contribution of shallow and deep features to the final result in the cross-feature dimension.
3. Extensive experiments verify that the ADWM module can be seamlessly integrated into various existing networks, enhancing their performance and achieving state-of-the-art results in a plug-and-play manner.

2. Method

In this section, we first introduce CACW, followed by an overview of ADWM, a detailed explanation of the proposed IFW and CFW, and a supplement on the plug-and-play integration of ADWM.

2.1. Correlation-Aware Covariance Weighting

In this section, we will introduce the design of CACW. Fig. 3 illustrates the overall process of CACW.

Background of PCA [1]. PCA is a common technique for dimensionality reduction. Let $X \in \mathbb{R}^{m\times n}$ represent the input observation matrix, where m denotes the number of samples and n denotes the number of features to be explored. The covariance matrix $C \in \mathbb{R}^{n\times n}$ can be calculated as follows:

$$C = \frac{1}{m-1}(X-\bar{X})^T(X-\bar{X}), \tag{1}$$

where $\bar{X}$ is the mean of each feature in X. Next, we perform eigenvalue decomposition and select the top k eigenvectors $\mathbf{v}_i$ with the largest eigenvalues λ_i to form matrix $P = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$. The process of generating $\mathbf{v}_i$ can be formulated as follows:

$$C\mathbf{v}_i = \lambda_i \mathbf{v}_i. \tag{2}$$

By projecting X onto this subspace using P, we obtain a reduced-dimensional representation Y. The process of projecting is as follows:

$$Y = P^T X. \tag{3}$$

CACW. The purpose of CACW is to generate weights for feature selection through adaptive adjustment, rather than dimensionality reduction via projection. In PCA, eigenvectors $\mathbf{v}_i$ represent the main directions of variation, while in CACW, the weight generation process establishes a basis to highlight important features and suppress redundancy. We first compute the covariance matrix C using Eq. (1), capturing correlations that reflect feature heterogeneity and redundancy. To mitigate the influence of absolute magnitudes, we normalize C to obtain $\tilde{C}$. The process of normalization is as follows:

$$\tilde{C}_{ij} = \frac{C_{ij}}{\|\bar{X}_i\|\|\bar{X}_j\|}, \tag{4}$$

where $\|\bar{X}_i\|$ and $\|\bar{X}_j\|$ are the norms of the respective feature vectors and each element $\tilde{C}_{ij}$ represents the normalized similarity between features i and j. As shown in Eq. (2), PCA's eigenvalue decomposition relies on linear feature decomposition, which is not data-driven and cannot adaptively highlight important features while suppressing redundancy. To address this, we achieve linear feature decomposition through a neural network. Specifically, we use a multi-layer perceptron (MLP) g to nonlinearly map the covariance matrix $\tilde{C}$ to obtain the weight vector $\gamma \in \mathbb{R}^{n\times 1}$. The process of generating weights is as follows:

$$\gamma = g(\tilde{C}), \tag{5}$$

where γ can represent either channel-weight in IFW or layer-weight in CFW.

Difference with attention. CACW stems from the covariance-based observation in Fig. 2 and the motivation to reduce feature redundancy and enhance heterogeneity, rather than being inspired by attention. Due to the difference between inspiration and motivation, its operation also fundamentally differs from attention. The scaling normalizes the covariance matrix to a correlation matrix, guiding the MLP to capture correlations, while attention's scaling mainly addresses gradient stability. CACW's MLP is placed after the covariance matrix for PCA-like eigenvalue decomposition in a data-driven way, while attention's MLP comes before QKV to extract nonlinear features, ignoring data correlations.

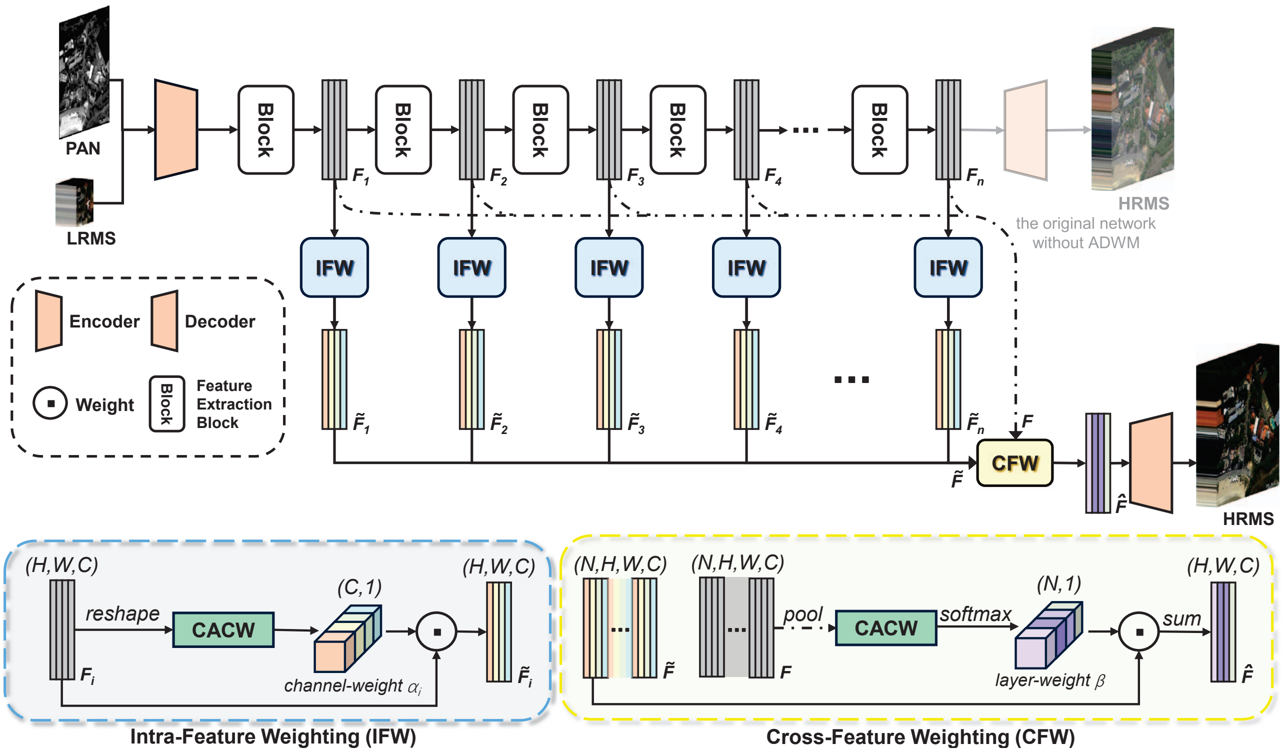

Figure 4. The overall workflow of ADWM is comprised of two sub-modules: Intra-Feature Weighting (IFW) and Cross-Feature Weighting (CFW). In IFW, each original feature F_i is adjusted to $\tilde{F}_i$ based on its internal correlations. In CFW, weights are generated based on the correlations among F_i features, dynamically adjusting each $\tilde{F}_i$'s contribution to the final output.

2.2. Overview of ADWM

We denote PAN as $P \in \mathbb{R}^{H\times W}$, LRMS as $L \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times c}$, and HRMS as $H \in \mathbb{R}^{H\times W\times c}$. In many methods, P and L are processed through an encoder and then fed into a sequential feature extraction block, such as Resblock [17], generating multiple intermediate features $F_i \in \mathbb{R}^{H\times W\times C}$, where i represents the i-th layer. Our ADWM is applied to adjust these features, maintaining generality and flexibility by integrating directly into existing network architectures without requiring modifications to the original structure. Firstly, each feature F_i is adaptively weighted based on its own correlation, resulting in $\tilde{F}_i \in \mathbb{R}^{H\times W\times C}$. Then, we collect F_i and $\tilde{F}_i$, $i = 1, 2, \ldots, n$ to obtain F and $\tilde{F} \in \mathbb{R}^{N\times H\times W\times C}$. We generate weights based on F and apply them to $\tilde{F}$, resulting in the adjusted $\hat{F}$. The dual-level weighting process can be formulated as follows:

$$\tilde{F}_i = \mathrm{IFW}(F_i), \tag{6}$$

$$F = [F_1, \ldots, F_n], \quad \tilde{F} = [\tilde{F}_1, \ldots, \tilde{F}_n], \tag{7}$$

$$\hat{F} = \mathrm{CFW}(F, \tilde{F}), \tag{8}$$

where IFW and CFW represent the intra-feature and cross-feature weighting processes, respectively, which are detailed in Sec. 2.3 and Sec. 2.4. $\hat{F}$ is then processed by a decoder to obtain the final output H.

2.3. Intra-Feature Weighting

The details of IFW: First, we reshape F_i to obtain $F_i^R \in \mathbb{R}^{HW\times C}$, focusing on exploring the correlations between channels by treating the HW spatial pixels as our sample space. Second, to dynamically adjust each feature F_i, we calculate the channel-weight $\alpha_i \in \mathbb{R}^{C\times 1}$, which addresses the heterogeneity and redundancy within the feature by assigning individualized importance to each channel based on its unique contribution. The process of generating channel-weight α_i is as follows:

$$F_i^R = \mathrm{Reshape}(F_i), \tag{9}$$

$$\alpha_i = f(F_i^R), \tag{10}$$

where $f(\cdot)$ denotes the CACW. The channel-weight α_i is then applied to F_i, adjusting the relative importance of each channel to produce the adjusted feature $\tilde{F}_i$. The weighting process is as follows:

$$\tilde{F}_i = F_i \odot \alpha_i, \tag{11}$$

where $\odot$ denotes element-wise production. Unlike the PCA's global orthogonal projection in Eq. (3), IFW independently

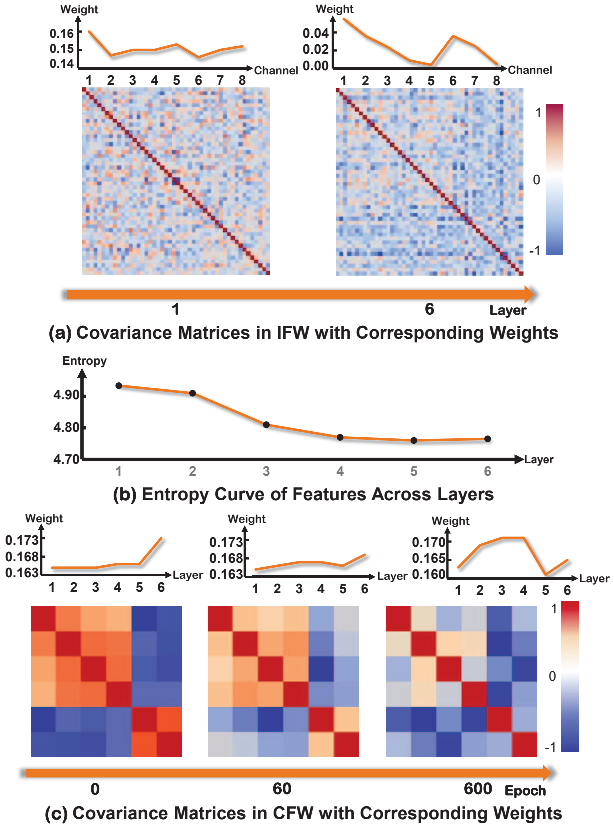

Figure 5. Visualization of covariance matrices, weights in IFW and CFW. (a) Channels that are multiples of six selected for clarity. (b) Lower entropy indicates higher feature redundancy.

scales each feature dimension through the generated weights. This can be seen as a simplified form of projection that preserves the original feature basis while optimizing the distribution of information.

The impact of IFW: As shown in Fig. 5 (a), shallower layers have lighter matrices, reflecting greater information diversity, while deeper layers darken, indicating high redundancy, as confirmed by the decreasing entropy trend in Fig. 5 (b). IFW adjusts weights accordingly, producing varying distributions across layers. In shallow layers, weights are uniform (0.141–0.155), while in deeper layers, they vary more (0.004–0.055), emphasizing key channels and suppressing others, helping the model focus on critical structures.

2.4. Cross-Feature Weighting

The details of CFW: The goal of CFW is to use adaptive weighting to fully leverage all intermediate features, effectively addressing heterogeneity and redundancy across layers. First, we apply spatial average pooling to adjust the shape of F, resulting in a representation $F^P \in \mathbb{R}^{C\times N}$, which enables us to explore the correlations between the N layers by treating the C channels as our sample space. Second, to dynamically adjust the contribution of different layer features

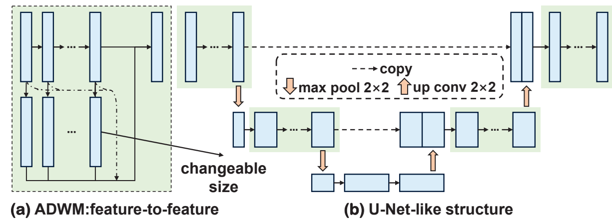

Figure 6. (a) ADWM can be seen as a feature-to-feature method. (b) The way it is embedded into more complex networks.

to the final result, we calculate the layer-weight $\beta \in \mathbb{R}^{N\times 1}$. The process of generating channel-weight β can be formulated as follows:

$$F^P = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{ij}, \tag{12}$$

$$\beta = f(F^P), \tag{13}$$

where the reshape operation flattens the width and height of the original input and $f(\cdot)$ denotes the CACW. Unlike IFW, CFW produces a single output feature $\hat{F}$ synthesized from N intermediate features. First, we use softmax on β to normalize layer weights, balancing depth contributions. Then, these weights modulate each intermediate feature $\tilde{F}$ reducing redundancy by giving less weight to repetitive layers. A final summation produces the integrated feature $\hat{F} \in \mathbb{R}^{H\times W\times C}$. The weighting process can be formulated as follows:

$$\hat{F} = \sum_{k=1}^{N}\left(\tilde{F}\odot \text{softmax}(\beta)\right), \tag{14}$$

where $\odot$ denotes element-wise production, and $\sum_{k=1}^{N}$ represents the summation across the N-dimension. The pointwise weighting and summation process in Eq. (14) can also be rewritten in a matrix multiplication form as follows:

$$\hat{F} = (\text{softmax}(\beta))^T \tilde{F}. \tag{15}$$

This corresponds to Eq. (3) in form. However, unlike PCA's global dimensionality reduction using a fixed orthogonal basis, CFW dynamically learns and applies task-specific weights β.

The impact of CFW: As shown in Fig. 5 (c), the CFW covariance matrix evolves during training, transitioning from deep red and blue regions to an overall lighter color, indicating reduced redundancy and increased feature diversity. As training progresses, the layer weights generated by CFW adjust gradually, reflecting the model's adaptive tuning of each layer's impact. These shifts reflect the model's refinement to align layer contributions with task demands, enhancing its ability to leverage diverse features and optimize performance throughout learning.

2.5. Flexible Plug-and-Play Integration

In addition to being integrated with the entire network in the way shown in Fig. 4, ADWM can also be flexibly combined

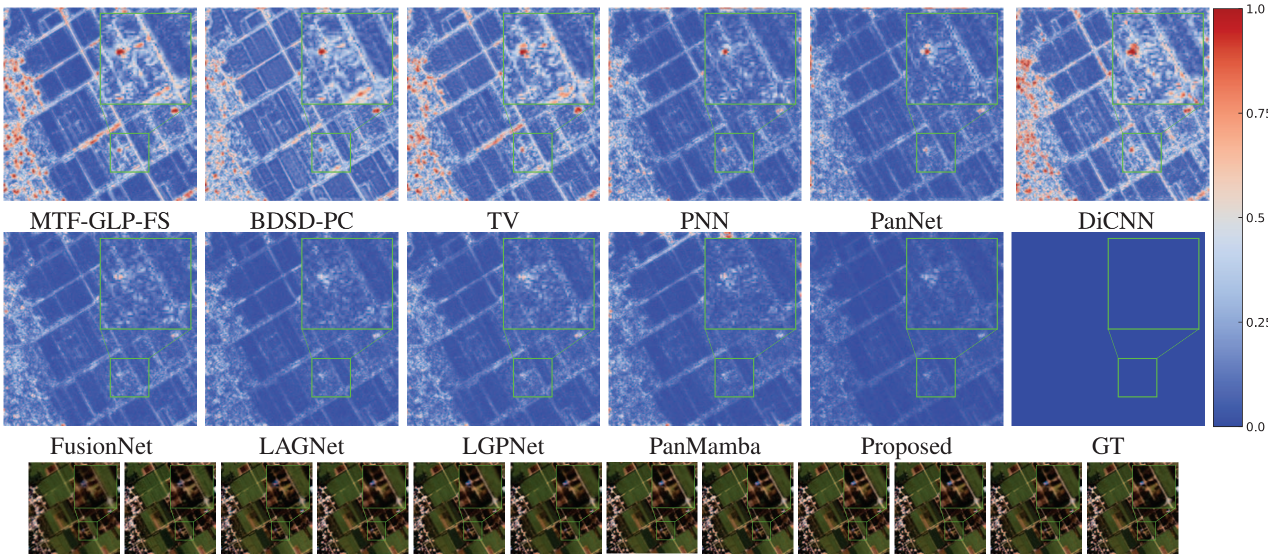

Figure 7. The residuals (Top) and visual results (bottom) of all compared approaches on the GF2 reduced-resolution dataset.

Table 1. Comparisons on WV3, QB, and GF2 reduced-resolution datasets, each with 20 samples. Best:**bold** , and second-best: underline.

Methods	WV3				QB				GF2			
	PSNR↑	SAM↓	ERGAS↓	Q8↑	PSNR↑	SAM↓	ERGAS↓	Q4↑	PSNR↑	SAM↓	ERGAS↓	Q4↑
MTF-GLP-FS [35]	32.963	5.316	4.700	0.833	32.709	7.792	7.373	0.835	35.540	1.655	1.589	0.897
BDSD-PC [33]	32.970	5.428	4.697	0.829	32.550	8.085	7.513	0.831	35.180	1.681	1.667	0.892
TV [30]	32.381	5.692	4.855	0.795	32.136	7.510	7.690	0.821	35.237	1.911	1.737	0.907
PNN [29]	37.313	3.677	2.681	0.893	36.942	5.181	4.468	0.918	39.071	1.048	1.057	0.960
PanNet [40]	37.346	3.613	2.664	0.891	34.678	5.767	5.859	0.885	40.243	0.997	0.919	0.967
DiCNN [18]	37.390	3.592	2.672	0.900	35.781	5.367	5.133	0.904	38.906	1.053	1.081	0.959
FusionNet [10]	38.047	3.324	2.465	0.904	37.540	4.904	4.156	0.925	39.639	0.974	0.988	0.964
LAGNet [24]	38.592	3.103	2.291	0.910	38.209	4.534	3.812	0.934	42.735	0.786	0.687	0.980
LGPNet [43]	38.147	3.270	2.422	0.902	36.443	4.954	4.777	0.915	41.843	0.845	0.765	0.976
PanMamba [19]	39.012	**2.913**	2.184	0.920	37.356	4.625	4.277	0.929	42.907	0.743	0.684	0.982
Proposed	**39.170**	**2.913**	**2.145**	**0.921**	**38.466**	**4.450**	**3.705**	**0.937**	**43.884**	**0.672**	**0.597**	**0.985**

with more complex networks. As shown in Fig. 6 (a), it can be seen as a feature-to-feature method, deriving the next feature from a series of sequential features. Especially, as shown in Fig. 6 (b), complex networks like U-Net often consist of parts with similar feature sizes and semantics, and an independent ADWM is applied to each part. As shown in Tab. 2, it still improves performance even when applied to a more complex network.

3. Experiments

3.1. Datasets, Metrics, and Training Details

In our experiments, we employ three datasets derived from satellite imagery captured by WorldView-3 (WV3), QuickBird (QB), and GaoFen-2 (GF2), constructed in accordance with Wald's protocol [37]. The datasets and associated data processing techniques are obtained from the PanCollection repository[1] [11]. We use well-established evaluation metrics to assess our method. For the reduced-resolution dataset, we employ SAM [5], ERGAS [36], Q4/Q8 [15], and PSNR. For the full-resolution dataset, D_s, D_λ, and HQNR [2] are used as evaluation metrics. Among them, HQNR is derived from D_s and D_λ, providing a comprehensive assessment of image quality. In addition, we train our model in Tab. 1 using the ℓ_1 loss function and the Adam optimizer [25] with a batch size of 64. The training details of our method in Tab. 2 remain consistent with those in the original paper. All experiments are conducted on an NVIDIA GeForce GTX 3090 GPU. *More details can be found in supplementary materials.*

3.2. Comparison with SOTA methods

We evaluate our method against nine competitive approaches, including 1) three classical methods: MTF-GLP-FS [35],

[1] https://github.com/liangjiandeng/PanCollection

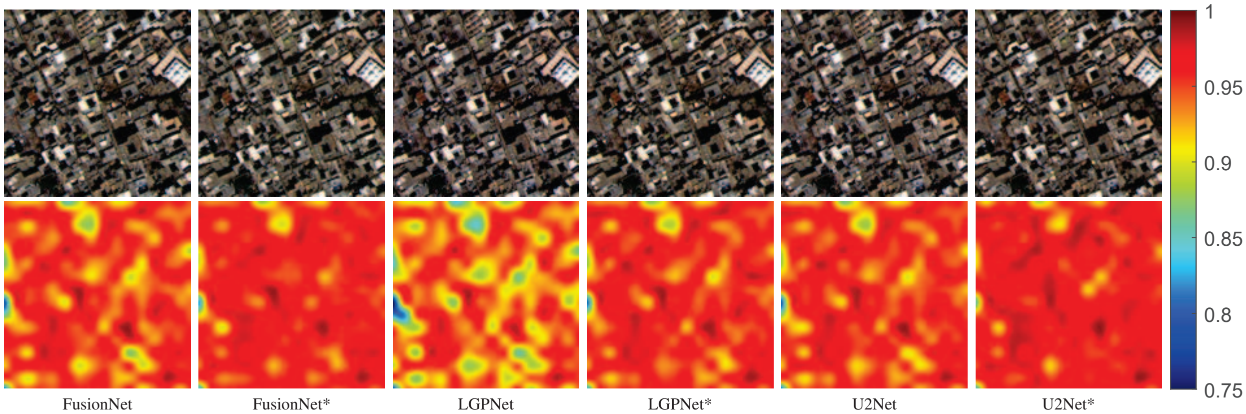

Figure 8. The visual results (Top) and HQNR maps (Bottom) of all evaluated general methods on the WV3 full-resolution dataset.

Table 2. Comparisons on WV3, QB, and GF2 datasets with 20 full-resolution samples, respectively. Methods marked with * represent the corresponding method enhanced with our ADWM module without any further changes. The best results in each column are **bolded**.

Method	Params	WV3			QB			GF2		
		$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑
FusionNet [10]	78.6K	0.024	0.037	0.940	**0.057**	0.052	0.894	0.035	0.101	0.867
FusionNet*	85.5K	**0.022**	**0.033**	**0.946**	0.066	**0.038**	**0.899**	**0.034**	**0.094**	**0.875**
LGPNet [43]	27.0K	0.022	0.039	0.940	0.093	0.061	0.870	0.030	0.081	0.892
LGPNet*	33.4K	**0.020**	**0.032**	**0.950**	**0.068**	**0.041**	**0.894**	**0.027**	**0.068**	**0.908**
U2Net [31]	757.5K	0.020	0.028	0.952	**0.052**	0.037	0.913	0.024	0.051	0.927
U2Net*	781.4K	**0.019**	**0.026**	**0.955**	0.054	**0.029**	**0.919**	**0.021**	**0.049**	**0.931**

BDSD-PC [33], and TV [30]; 2) seven deep learning-based methods PNN [29], PanNet [40], DiCNN [18], FusionNet [10], LAGNet [24], LGPNet [43], and PanMamba [19]. We incorporated ADWM into the classic LAGNet as our proposed method. Tab. 1 presents a comprehensive comparison of our method with state-of-the-art approaches across three datasets. Notably, our method achieves this high level of performance across all metrics. Specifically, our method achieves a PSNR improvement of 0.158dB, 0.255dB, and 0.598dB on the WV3, QB, and GF2 datasets, respectively, compared to the second-best results. These improvements highlight the clear advantages of our method, confirming its competitiveness in the field. Fig. 7 provides qualitative assessment results for GF2 datasets alongside the respective ground truth (GT). By comparing the mean squared error (MSE) residuals between the pansharpened results and the ground truth, it is clear that our residual maps are the darkest, suggesting that our method achieves the highest fidelity among the evaluated methods.

3.3. Generality Experiment

Our ADWM serves as a plug-and-play mechanism, allowing it to be easily integrated into various existing frameworks without requiring extensive modifications. To further demonstrate the generality of our method, we integrated three different approaches into our ADWM framework, including FusionNet [10], LGPNet [43], and U2Net [31]. As shown in Tab. 2, our method improves performance for each approach across three datasets, with only a negligible increase in parameter size. In terms of visual quality shown in Fig. 8, as depicted in the second row, the redder areas indicate better performance, while the bluer areas indicate poorer performance. After integrating ADWM into all the methods, they all show larger and deeper red areas, indicating better performance. These results validate our method's significant potential for practical applications.

3.4. Redundancy Visualization

To demonstrate that our method improves by our motivation to reduce feature redundancy and enhance heterogeneity, we visualized the results using LGPNet [43] on the WV3 reduced-resolution dataset. We applied SVD to the covariance matrix of each intermediate feature, sorted the eigenvalues to obtain the corresponding scree plot [44], and averaged all the intermediate features to generate Fig. 9. Compared to the original network and self-attention-based weighting, our method yields the smoothest curve, indicating a more balanced distribution of variance across multiple dimensions. This suggests that information is more evenly spread rather than being concentrated in a few dominant components, leading to reduced redundancy. This reduction leads to higher performance metrics, such as PSNR.

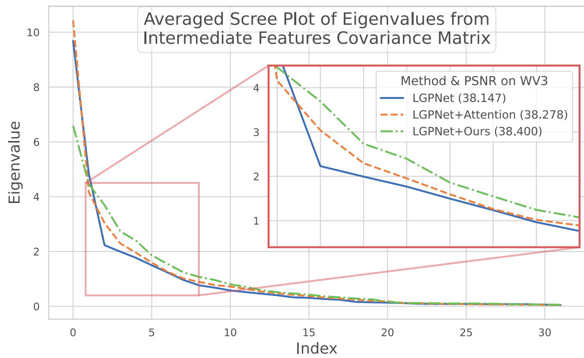

Figure 9. Scree Plot to illustrate the differences in redundancy. The smoother the curve, the lower the redundancy.

Table 3. Comparison of different weight generation approaches.

Method	Params	WV3			
		PSNR↑	SAM↓	ERGAS↓	Q8↑
PCA	11.2K	38.612	3.076	2.273	0.913
Pool	39.2K	39.053	2.915	2.183	0.920
Attention	42.5K	38.875	3.012	2.237	0.915
CACW	26.2K	**39.170**	**2.913**	**2.145**	**0.921**

3.5. Evaluation and Analysis of CACW

Comparison of Weighting Approaches: This section shows the comparison results of CACW with other commonly used approaches for generating weights. In the first row of Tab. 3, PCA selects the top half eigenvectors corresponding to the largest eigenvalues, which are then processed by an MLP to generate the weight. CACW outperforms PCA, showing that a neural network with nonlinear capabilities creates a more effective feature importance representation. In the second row, Pool denotes global average pooling, compressing each channel to a scalar and generating weights through an MLP. CACW surpasses Pool with fewer parameters by preserving inter-channel correlations via a covariance matrix, addressing the information loss and dependency limitations of pooling. In the third row, Attention uses self-attention to compute a relevance matrix before generating weights through an MLP. With fewer parameters, CACW outperforms Attention, avoiding the complexity and noise of attention mechanisms.

Impact of Intermediate Layer Size: This section explores the impact of the intermediate layer size d, a key variable in our method. Experiments were conducted using LAGNet [24] as the backbone on the WV3 reduced-resolution dataset. In this analysis, d is varied in IFW while keeping CFW fixed, with its FLOPs constant at 90, contributing only a small portion to the total computation. As shown in Fig. 10, the overall FLOPs increase as d becomes larger. In terms of performance, when d is too small, the network struggles to capture complex feature correlations, resulting in lower PSNR values. As d increases, PSNR improves, indicating enhanced image quality. However, when d exceeds $0.8n$, the network becomes overly complex, leading to overfitting and a subsequent drop in PSNR.

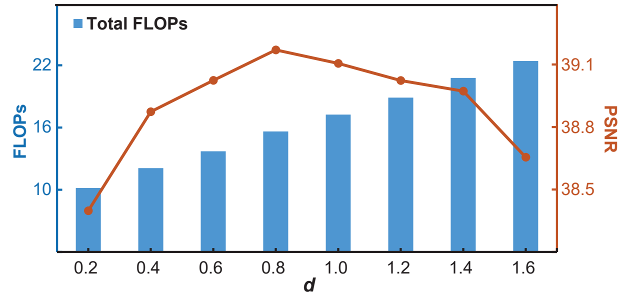

Figure 10. FLOPs (K) and PSNR (dB) with d represented as a fraction of n, both are variables in Fig. 3.

Table 4. Ablation experiment on WV3 reduced-resolution dataset.

IFW	CFW	WV3			
		PSNR↑	SAM↓	ERGAS↓	Q8↑
✗	✗	38.592	3.103	2.291	0.910
✓	✗	39.028	2.972	2.183	0.917
✗	✓	38.923	2.990	2.210	0.918
✓	✓	**39.170**	**2.913**	**2.145**	**0.921**

Complexity Analysis: The additional computational complexity of ADWM, compared to the original network, mainly stems from generating the covariance matrix. IFW has a theoretical complexity of $O(C^2 \cdot (H \cdot W))$, where H and W are the spatial dimensions, and C is the number of channels. CFW has a theoretical complexity of $O(N^2 \cdot C)$, where N is the number of feature maps being processed. In total, the complexity of our module is $O(N \cdot (H \cdot W) \cdot C^2 + N^2 \cdot C)$, indicating that the main computational cost is driven by the number of channels C and the spatial dimensions H and W.

3.6. Ablation Study

We conducted ablation studies with LAGNet [24] on the WV3 dataset. In the second row of Tab. 4, replacing CFW's dynamic weighting with equal weights causes a performance drop, highlighting the importance of dynamic adjustment. In the third row, omitting IFW caused a significant decline, underscoring its role in handling feature heterogeneity and redundancy. However, performance still exceeded the first row, demonstrating CFW's effectiveness in integrating shallow and deep features.

4. Conclusion

In this paper, we use the covariance matrix to model feature heterogeneity and redundancy, introducing CACW to capture these correlations and generate weights for effective adjustment. Building on CACW, we propose the adaptive dual-level weighting mechanism (ADWM), which includes Intra-Feature Weighting (IFW) and Cross-Feature Weighting (CFW). ADWM significantly enhances a wide range of existing deep learning methods, and its effectiveness is thoroughly demonstrated through extensive experiments.

Acknowledgement: This work was supported in part by the NSFC (12271083, 62273077), Sichuan Province's Science and Technology Empowerment for Disaster Prevention, Mitigation, and Relief Project (2025YFNH0001), and the Natural Science Foundation of Sichuan Province (2024NSFJQ0013).

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 2, 3

[2] Alberto Arienzo, Gemine Vivone, Andrea Garzelli, Luciano Alparone, and Jocelyn Chanussot. Full-resolution quality assessment of pansharpening: Theoretical and hands-on approaches. *IEEE Geoscience and Remote Sensing Magazine*, 10(3):168–201, 2022. 6

[3] Anju Asokan and JJESI Anitha. Change detection techniques for remote sensing applications: a survey. *Earth Science Informatics*, 12(2):143–160, 2019. 1

[4] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):354–379, 2012. 1

[5] Joseph W Boardman. Automating spectral unmixing of aviris data using convex geometry concepts. In *JPL, Summaries of the 4th Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1993. 6

[6] Xiangyong Cao, Feng Zhou, Lin Xu, Deyu Meng, Zongben Xu, and John Paisley. Hyperspectral image classification with markov random fields and a convolutional neural network. *IEEE Transactions on Image Processing*, 27(5):2354–2367, 2018. 1

[7] Xiangyong Cao, Jing Yao, Zongben Xu, and Deyu Meng. Hyperspectral image classification with convolutional neural network and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(7):4604–4616, 2020. 1

[8] Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11–28, 2016. 1

[9] Jaewan Choi, Kiyun Yu, and Yongil Kim. A New Adaptive Component-Substitution-Based Satellite Image Fusion by Using Partial Replacement. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1):295–309, 2011. 2

[10] Liang-Jian Deng, Gemine Vivone, Cheng Jin, and Jocelyn Chanussot. Detail injection-based deep convolutional neural networks for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, page 6995–7010, 2021. 6, 7

[11] Liang-jian Deng, Gemine Vivone, Mercedes E. Paoletti, Giuseppe Scarpa, Jiang He, Yongjun Zhang, Jocelyn Chanussot, and Antonio Plaza. Machine Learning in Pansharpening: A benchmark, from shallow to deep networks. *IEEE Geoscience and Remote Sensing Magazine*, 10(3):279–315, 2022. 6

[12] Renwei Dian, Shutao Li, Anjing Guo, and Leyuan Fang. Deep hyperspectral image sharpening. *IEEE transactions on neural networks and learning systems*, 29(11):5345–5355, 2018. 2

[13] Xueyang Fu, Zihuang Lin, Yue Huang, and Xinghao Ding. A Variational Pan-Sharpening With Local Gradient Constraints. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10257–10266. IEEE, 2019. 2

[14] Xueyang Fu, Wu Wang, Yue Huang, Xinghao Ding, and John Paisley. Deep multiscale detail networks for multiband spectral image sharpening. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2090–2104, 2021. 2

[15] Andrea Garzelli and Filippo Nencini. Hypercomplex Quality Assessment of Multi/Hyperspectral Images. *IEEE Geoscience and Remote Sensing Letters*, 6(4):662–665, 2009. 6

[16] Hassan Ghassemian. A review of remote sensing image fusion methods. *Information Fusion*, 32:75–89, 2016. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[18] 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. 6, 7

[19] Xuanhua He, Ke Cao, Ke Ren Yan, Rui Li, Chengjun Xie, Jie Zhang, and Man Zhou. Pan-mamba: Effective pan-sharpening with state space model. *ArXiv*, abs/2402.12192, 2024. 6, 7

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 2

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[22] Jie Huang, Rui Huang, Jinghao Xu, Siran Pen, Yule Duan, and Liangjian Deng. Wavelet-assisted multi-frequency attention network for pansharpening. *arXiv preprint arXiv:2502.04903*, 2025. 2

[23] Gong Jianya, Sui Haigang, Ma Guorui, and Zhou Qiming. A review of multi-temporal remote sensing data change detection algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (B7):757–762, 2008. 1

[24] Zi-Rong Jin, Tian-Jing Zhang, Tai-Xiang Jiang, Gemine Vivone, and Liang-Jian Deng. Lagconv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1113–1121, 2022. 6, 7, 8

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[26] Shutao Li, Renwei Dian, Leyuan Fang, and José M. Bioucas-Dias. Fusing hyperspectral and multispectral images via coupled sparse tensor factorization. *IEEE Transactions on Image Processing*, 27(8):4118–4130, 2018. 2

[27] Hefei Ling, Jiyang Wu, Lei Wu, Junrui Huang, Jiazhong Chen, and Ping Li. Self residual attention network for deep face recognition. *IEEE Access*, 7:55159–55168, 2019. 2

[28] Junmin Liu, Shijie Li, Changsheng Zhou, Xiangyong Cao, Yong Gao, and Bo Wang. Sraf-net: A scene-relevant anchor-free object detection network in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/TGRS.2021.3124959, 2021. 1

[29] Giuseppe Masi, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7):594, 2016. 6, 7

[30] Frosti Palsson, Johannes R. Sveinsson, and Magnus O. Ulfarsson. A new pansharpening algorithm based on total variation. *IEEE Geoscience and Remote Sensing Letters*, page 318–322, 2013. 6, 7

[31] Siran Peng, Chenhao Guo, Xiao Wu, and Liang-Jian Deng. U2net: A general framework with spatial-spectral-integrated double u-net for image fusion. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*, pages 3219–3227, 2023. 7

[32] Xin Tian, Yuerong Chen, Changcai Yang, and Jiayi Ma. Variational Pansharpening by Exploiting Cartoon-Texture Similarities. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022. 2

[33] Gemine Vivone. Robust band-dependent spatial-detail approaches for panchromatic sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, page 6421–6433, 2019. 2, 6, 7

[34] Gemine Vivone, Rocco Restaino, Mauro Dalla Mura, Giorgio Licciardi, and Jocelyn Chanussot. Contrast and Error-Based Fusion Schemes for Multispectral Image Pansharpening. *IEEE Geoscience and Remote Sensing Letters*, 11(5): 930–934, 2014. 2

[35] Gemine Vivone, Rocco Restaino, and Jocelyn Chanussot. Full scale regression-based injection coefficients for panchromatic sharpening. *IEEE Transactions on Image Processing*, 27(7): 3418–3431, 2018. 2, 6

[36] Lucien Wald. *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*. Presses des MINES, 2002. 6

[37] Lucien Wald, Thierry Ranchin, and Marc Mangolini. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric engineering and remote sensing*, 63(6):691–699, 1997. 6

[38] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 2

[39] Xinyu Xie, Yawen Cui, Tao Tan, Xubin Zheng, and Zitong Yu. Fusionmamba: Dynamic feature enhancement for multimodal image fusion with mamba. *Visual Intelligence*, 2(1):37, 2024. 2

[40] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1753–1761, 2017. 6, 7

[41] Yong Yang, Lei Wu, Shuying Huang, Weiguo Wan, Wei Tu, and Hangyuan Lu. Multiband remote sensing image pansharpening based on dual-injection model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:1888–1904, 2020. 2

[42] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 2

[43] Chen-Yu Zhao, Tian-Jing Zhang, Ran Ran, Zhi-Xuan Chen, and Liang-Jian Deng. Lgpconv: Learnable gaussian perturbation convolution for lightweight pansharpening. In *IJCAI*, pages 4647–4655, 2023. 6, 7

[44] Mu Zhu and Ali Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006. 7

[45] Peixian Zhuang, Qingshan Liu, and Xinghao Ding. Pan-ggf: A probabilistic method for pan-sharpening with gradient domain guided image filtering. *Signal Processing*, 156:177–190, 2019. 2