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

Supplementary Material

Abstract

The supplementary materials provide additional insights into the ADWM proposed in our paper. We present an in-depth exploration of how our method leverages covariance matrices and draws connections to PCA, highlighting its distinctions from prior approaches. Key variables influencing performance are thoroughly examined. The implementation approach, datasets, and training details are also provided. Finally, we present additional experimental results on visual analysis, comparison with SOTA methods, and the generality experiment. Code will be provided after acceptance.

A. Analysis on Covariance Matrix and PCA

Many prior works [5, 7, 14] have explored adaptive weighting to enhance feature representation. Our method introduces two key innovations: 1) leveraging covariance matrix correlations to explicitly capture feature heterogeneity and redundancy; and 2) both IFW and CFW have conceptual links to PCA, providing stronger theoretical support.

Covariance matrix. Covariance matrices are symmetric, with off-diagonal elements C_{ij} denoting the covariance between features i and j. High absolute covariance values indicate strong linear correlations, while low values suggest weaker dependencies. In our method, covariance captures both feature redundancy and heterogeneity. High covariance often implies redundancy, as features convey overlapping information. For example, two features strongly correlated with shared attributes (e.g., brightness or texture) result in high covariance. In contrast, low covariance reflects heterogeneity, where features provide distinct information, such as spectral properties versus spatial textures. This approach offers a significant advantage over traditional weighting mechanisms, which rely on simple global statistics or local operations and fail to model complex global relationships among features. By explicitly modeling these relationships using the covariance matrix, *our method inherently incorporates non-local properties, quantifying the dependencies between features. As a result, weights are adaptively adjusted to enhance features with high heterogeneity, while suppressing those with high redundancy. This not only reduces redundant information but also significantly improves the overall informativeness of the feature representations.*

Link Between CFW and PCA. In PCA, the eigenvectors $\mathbf{v}_i$ represent the principal directions of variation in the data. Similarly, our CACW generates weights β that form a basis to highlight important features and suppress redundancy, providing a clear and interpretable theoretical foundation. The relationship between CFW and PCA is particularly notable, as it parallels PCA's projection operation, where the original data matrix X is transformed using the eigenvector matrix P, formulated as:

$$Y = P^T X, \tag{1}$$

where Y represents the reduced-dimensional data. The pointwise weighting and summation process in CFW can also be rewritten in a matrix multiplication form, which corresponds to Eq. (1) in structure. The weighting process in our method can be formulated as:

$$\hat{F} = (\text{softmax}(\beta))^T \tilde{F}, \tag{2}$$

where $\hat{F}$ represents the aggregated critical information from all intermediate results, effectively reducing redundancy and capturing the essence of the feature space. However, unlike PCA's global dimensionality reduction using a fixed orthogonal basis, CFW dynamically learns and applies task-specific weights β. This adaptive aggregation tailors feature representation to the task, enhancing relevant features and overall representation.

Link Between IFW and PCA. The IFW also demonstrates a conceptual link to PCA, though with a key distinction. Instead of performing a matrix multiplication for dimensionality reduction, IFW employs a channel-wise pointwise multiplication between the generated weights and the original features. The weighting process is formulated as follows:

$$\tilde{F}_i = F_i \odot \alpha_i, \tag{3}$$

where $\odot$ denotes element-wise multiplication. Unlike the global orthogonal projection in PCA represented by Eq. (1), IFW independently 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. By selectively amplifying important feature channels and suppressing redundant ones, IFW refines the feature representation without requiring a global basis transformation.

A.1. Visual Analysis

A.1.1. Intra-Feature Visualization

Change of Covariance Matrix: As shown in Fig. 1 (a), In shallower layers, the overall color of the matrix is lighter, indicating more diverse information across channels, but it becomes darker with increasing depth, reflecting high redundancy. Additionally, as shown in Fig. 1 (b), the entropy curve

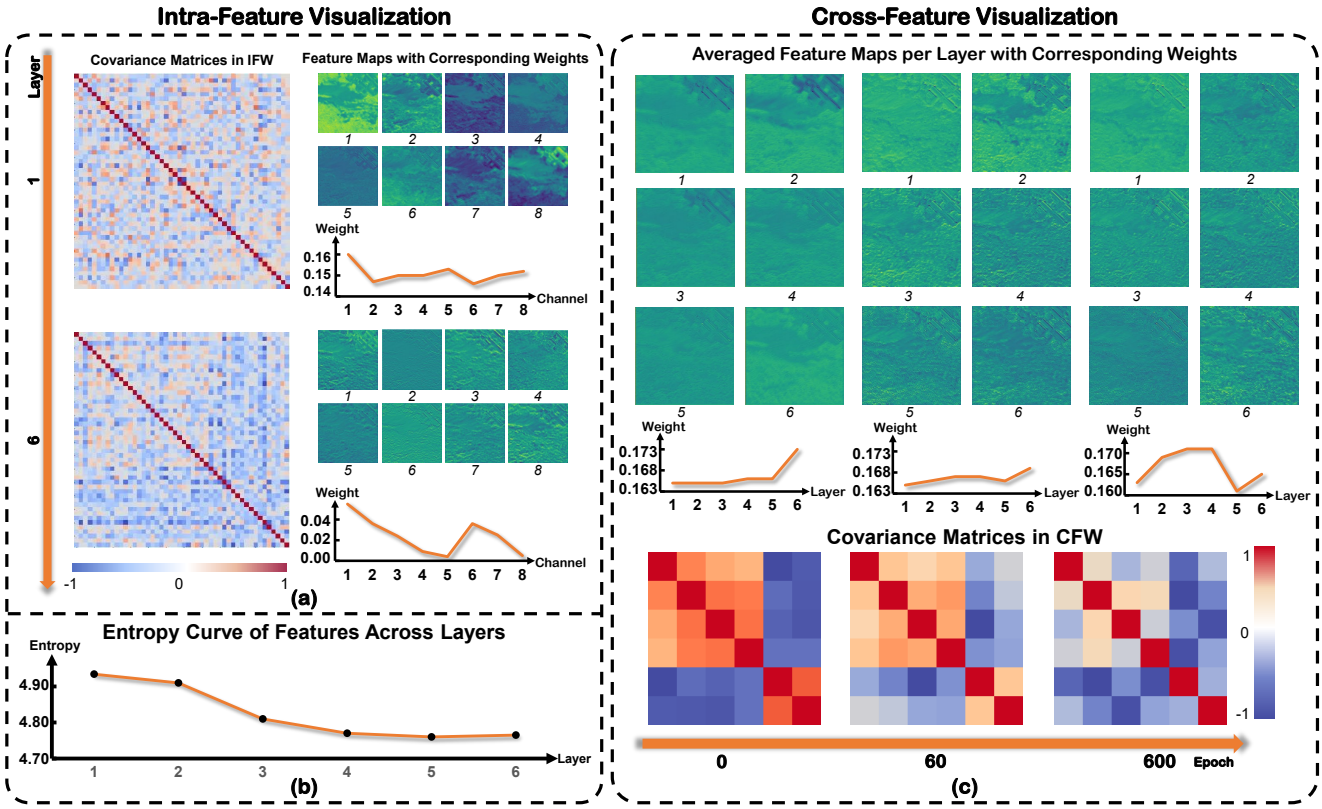

Figure 1. Visualization of covariance matrices, weights, and feature representations in IFW and CFW. In (a), intra-feature covariance matrices, channel weights, and corresponding feature maps across various layers are shown, with channels that are multiples of six selected for clarity. In (b), entropy variations of features across layers are displayed. In (c), cross-feature covariance matrices, layer weights, and average feature maps across different training epochs are illustrated.

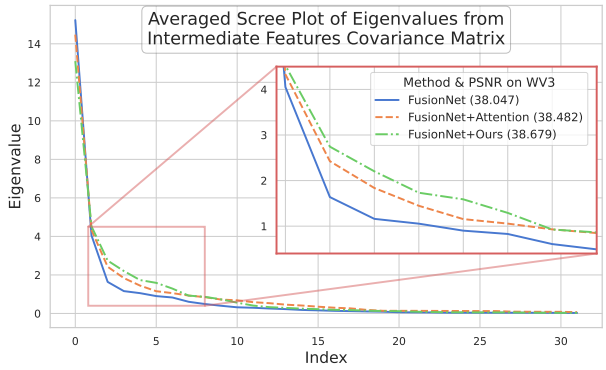

Figure 2. Scree Plot to illustrate the differences in redundancy. The smoother the curve, the lower the redundancy.

illustrates the level of information diversity across layers, with a decreasing trend indicating increased redundancy as the depth increases.

Changes of Channel Weight: The line charts within the feature maps illustrate the weights assigned to each channel. In the shallow layers, the weights are relatively uniform (0.141 to 0.155), indicating minimal differentiation among low-level features that do not require emphasis. In deeper layers, however, the weights show greater variance (0.004 to 0.055), with some channels emphasized and others minimized. The smaller weight values in deeper layers reflect the larger magnitude of deep-layer features. This distribution enables the model to capture foundational information in the shallow layers and selectively focus on important structural details in deeper layers.

Redundancy Visualization: To further demonstrate that our method improves by reducing feature redundancy and enhancing heterogeneity, we additionally visualized the results using FusionNet [1] on the WV3 reduced-resolution dataset. The analysis is the same as in Sec. 3.4 of the main text.

A.1.2. Cross-Feature Visualization

Change of Covariance Matrix: As illustrated in Fig. 1 (c), The covariance matrix generated by CFW shows significant evolution over the course of training. In the early epochs, the blue areas indicate negative correlations between shallow and deep features, while the red areas reveal positive correlations within the shallow and deep features themselves. Both types of correlations reflect redundancy in the feature representation. As training progresses, the overall color of the covariance matrix lightens, reflecting a gradual reduction in inter-feature correlation. This trend indicates increased

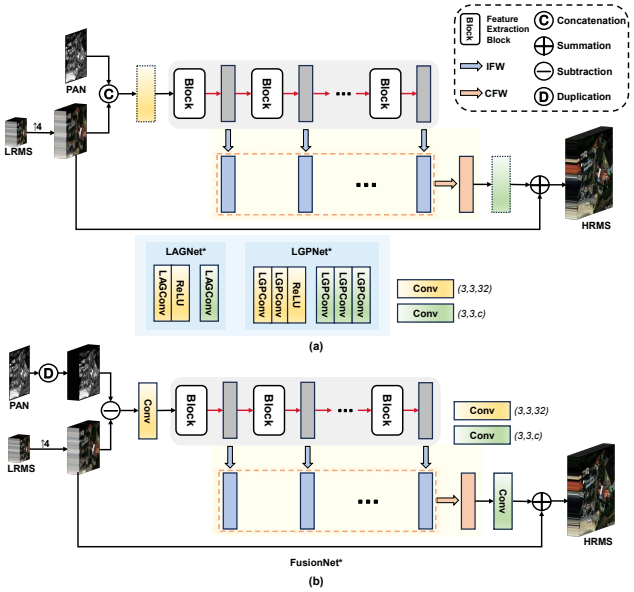

Figure 3. Illustrations of how our proposed module is integrated into various methods in a plug-and-play manner. (a) Integration into LAGNet [6] and LGPNet [16], which share the same framework but differ in their feature extraction blocks. The unique designs of the input and output stages are highlighted at the bottom. (b) Integration into FusionNet [1].

feature diversity and a decline in redundant information as training advances.

Changes of Layer Weight: As training progresses, layer weights adjust gradually, indicating 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.

B. Implementation Details of Plug-and-Play

This section demonstrates how our method can be seamlessly integrated into various existing approaches. Our ADWM serves as a plug-and-play mechanism. When comparing with SOTA methods, we incorporated the ADWM module into the classic LAGNet [6] as our proposed method. To further validate the generality of our approach, we integrated three baseline methods into our ADWM framework: FusionNet [1] and LGPNet [16]. The U2Net [10] has already been presented in the main text, Fig. 6. As shown in Fig. 3, all methods retain their original frameworks, with our dual-level weighting module integrated only into the intermediate continuous feature extraction blocks.

C. Datasets

We utilized datasets derived from the WorldView-3 (WV3), QuickBird (QB), and GaoFen-2 (GF2) satellites for our experiments. These datasets consist of image patches extracted from remote sensing imagery, which are separated into training and testing subsets. The training data includes image triplets of PAN, LRMS, and GT obtained through downsampling-based simulation, with respective dimensions of 64×64, $16 \times 16 \times C$, and $64 \times 64 \times C$. For the WV3 dataset, the training set contains approximately 10,000 samples with eight channels ($C = 8$). Similarly, the QB training set consists of about 17,000 samples with four channels ($C = 4$), while the GF2 dataset includes 20,000 samples with four channels ($C = 4$). The reduced-resolution test set for each satellite is composed of 20 PAN/LRMS/GT image triplets with a variety of representative land cover types. These images, simulated via downsampling, have dimensions of 256×256, $64 \times 64 \times C$, and $256 \times 256 \times C$, respectively. For the full-resolution testing phase, the dataset comprises 20 PAN/LRMS image pairs with sizes of 512×512 and 128×128. The datasets and processing procedures were obtained from the PanCollection repository [2].

D. Training Details

When comparing with SOTA methods, the training of ADWM on the WV3 dataset was conducted using the ℓ_1 loss function and the Adam optimizer. The batch size was set to 64, with an initial learning rate of 2×10^{-3}, decaying to half its value every 150 steps. The training process spanned 500 epochs. The network architecture used 48 channels in the hidden layers, and the intermediate layer size d in CACW was set to $0.8n$. For the QB dataset, the training process lasted 200 epochs, with the intermediate layer size d in CACW set to $0.6n$, while all other settings remained consistent with those used for the WV3 dataset. For the GF2 dataset, all settings were identical to the WV3 configuration. In general experiments, all other training settings followed those specified in the original papers.

E. Additional Results

Visual Analysis: To provide a more detailed illustration of the channel weights generated in IFW, we present the complete results of the first and sixth layers on one picture of GF2 reduced-resolution datasets, including all feature maps and their corresponding weights for each channel, as shown in Fig. 4.

Comparison with SOTA methods: Tab. 1 showcases a comprehensive comparison of our method with state-of-the-art approaches across three datasets on full-resolution images. The HQNR metric [13], which improves upon the QNR metric, evaluates both spatial and spectral consistency, offering a comprehensive reflection of the image-fusion effectiveness of different methods. It is widely regarded as one of the most important metrics for full-resolution datasets. Our method achieves the highest HQNR on all three datasets. Additionally, in Figs. 5 to 9, we provide visual comparisons of the

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

Methods	WV3			QB			GF2		
	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑
MTF-GLP-FS [12]	0.020	0.063	0.919	0.047	0.150	0.810	0.035	0.143	0.828
BDSD-PC [11]	0.063	0.073	0.870	0.198	0.164	0.672	0.076	0.155	0.781
TV [9]	0.023	0.039	0.938	0.055	0.100	0.850	0.055	0.112	0.839
PNN [8]	0.021	0.043	0.937	0.058	0.062	0.884	0.032	0.094	0.877
PanNet [15]	**0.017**	0.047	0.937	**0.043**	0.114	0.849	**0.018**	0.080	0.904
DiCNN [3]	0.036	0.046	0.920	0.095	0.107	0.809	0.037	0.099	0.868
FusionNet [1]	0.024	0.037	0.940	0.057	0.052	0.894	0.035	0.101	0.867
LAGNet [6]	0.037	0.042	0.923	0.086	0.068	0.852	0.028	0.079	0.895
LGPNet [16]	0.022	0.039	0.940	0.074	0.061	0.870	0.030	0.080	0.892
PanMamba [4]	0.018	0.053	0.930	0.049	0.044	0.910	0.023	0.057	0.921
Proposed	0.024	**0.029**	**0.948**	0.064	**0.024**	**0.914**	0.022	**0.052**	**0.928**

Table 2. Comparisons on WV3, QB, and GF2 datasets with 20 reduce-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	WV3				QB				GF2			
	PSNR↑	SAM↓	ERGAS↓	Q8↑	PSNR↑	SAM↓	ERGAS↓	Q4↑	PSNR↑	SAM↓	ERGAS↓	Q4↑
FusionNet [1]	38.047	3.324	2.465	0.904	37.540	4.904	4.156	0.925	39.639	0.974	0.988	0.964
FusionNet*	**38.679**	**3.097**	**2.249**	**0.909**	**38.271**	**4.654**	**3.795**	**0.933**	**41.649**	**0.832**	**0.769**	**0.975**
LGPNet [16]	38.147	3.270	2.422	**0.902**	36.443	4.954	4.777	0.915	41.843	0.845	0.765	0.976
LGPNet*	**38.400**	**3.241**	**2.330**	**0.902**	**38.309**	**4.666**	**3.773**	**0.932**	**42.230**	**0.828**	**0.714**	**0.978**
U2Net [10]	39.117	2.888	2.149	0.920	38.065	4.642	3.987	0.931	43.379	0.714	0.632	0.981
U2Net*	**39.234**	**2.864**	**2.143**	**0.922**	**38.762**	**4.432**	**3.691**	**0.939**	**43.901**	**0.670**	**0.592**	**0.986**

outputs generated by various methods on sample images from the WV3, QB, and GF2 datasets.

Generality Experiment: Tab. 2 presents the results of the generality experiment conducted across three datasets on reduced-resolution images. In Figs. 10 to 14, we showcase visual output comparisons for some methods before and after incorporating our ADWM module, using sample images from the WV3, QB, and GF2 datasets.

References

[1] 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. 2, 3, 4

[2] 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. 3

[3] 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. 4

[4] 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. 4

[5] 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. 1

[6] 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. 3, 4

[7] 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. 1

[8] Giuseppe Masi, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7):594, 2016. 4

[9] 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. 4

[10] 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. 3, 4

[11] Gemine Vivone. Robust band-dependent spatial-detail approaches for panchromatic sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, page 6421–6433, 2019. 4

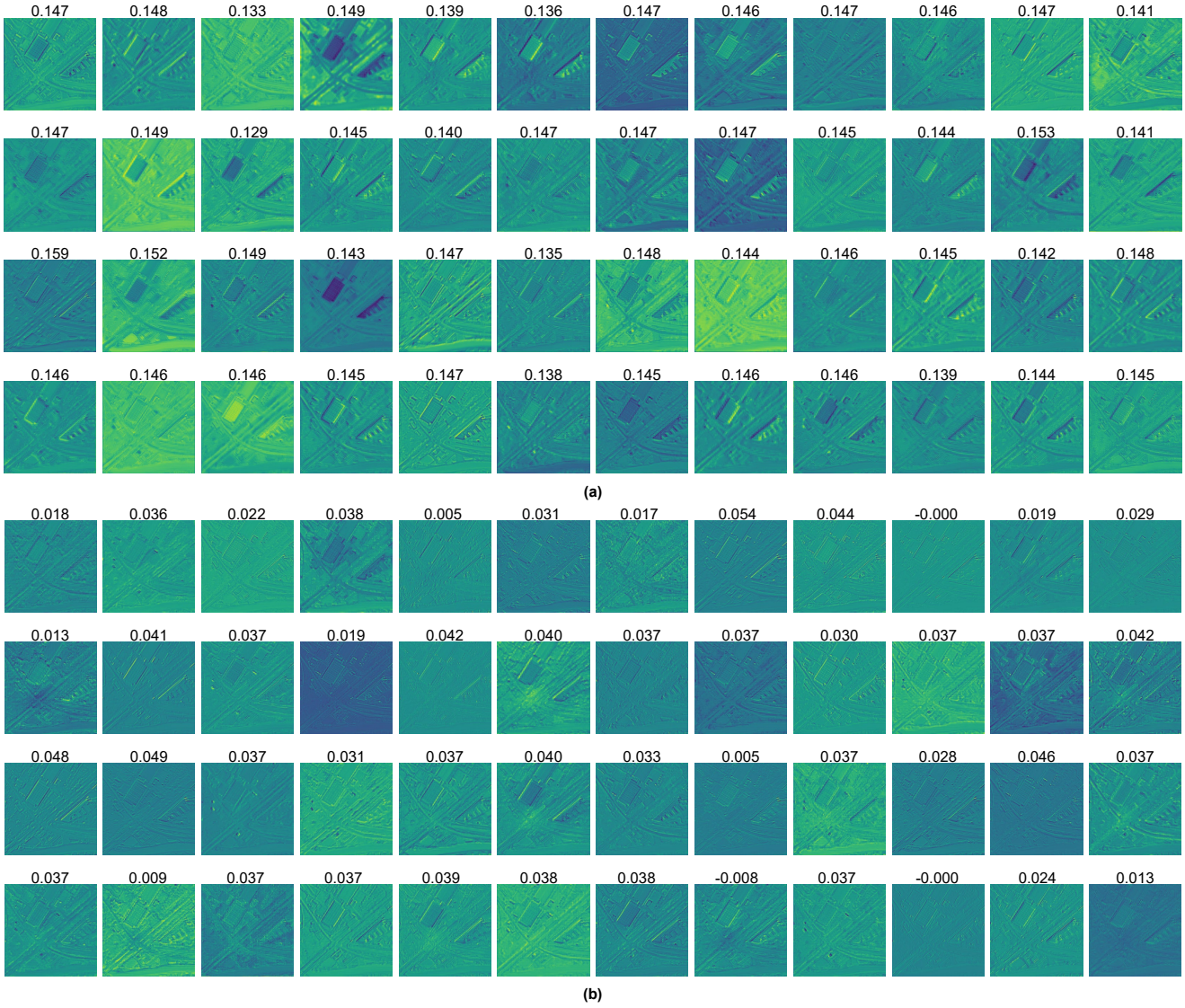

Figure 4. Channel weights and corresponding feature maps in IFW: (a) Results from the first layer, where each feature map is annotated with its corresponding weight. (b) Results from the sixth layer.

[12] 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. 4

[13] Gemine Vivone, Mauro Dalla Mura, Andrea Garzelli, Rocco Restaino, Giuseppe Scarpa, Magnus O. Ulfarsson, Luciano Alparone, and Jocelyn Chanussot. A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods. *IEEE Geoscience and Remote Sensing Magazine*, 9 (1):53–81, 2021. 3

[14] 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. 1

[15] 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. 4

[16] 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. 3, 4

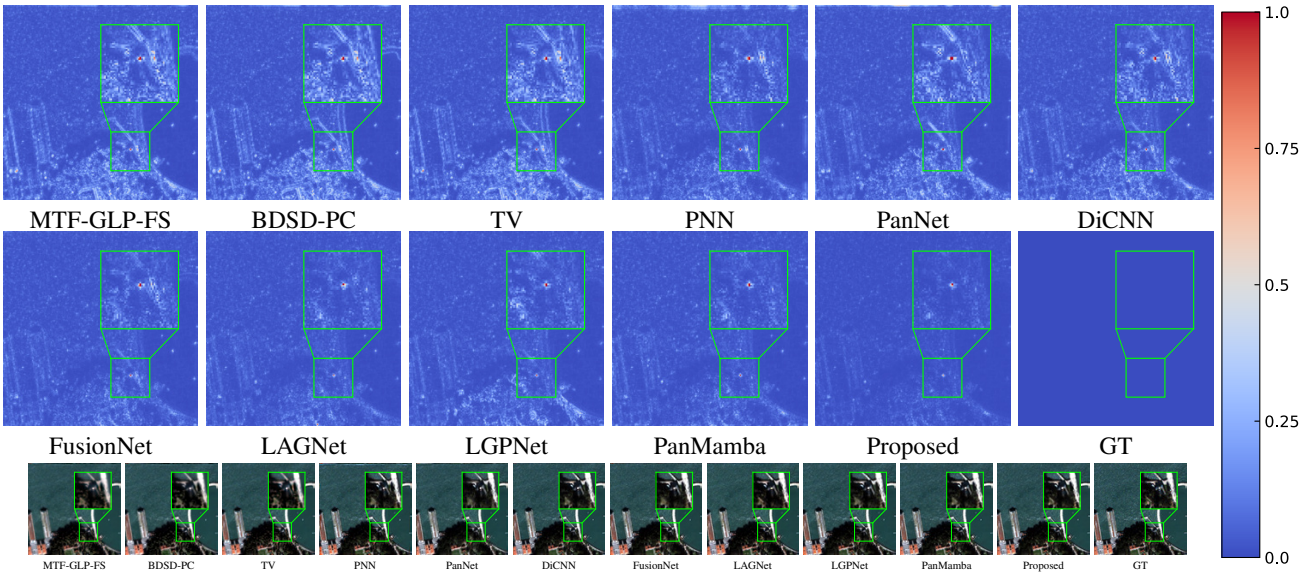

Figure 5. The residuals (Top) and visual results (bottom) of all compared approaches on the QB reduced-resolution dataset.

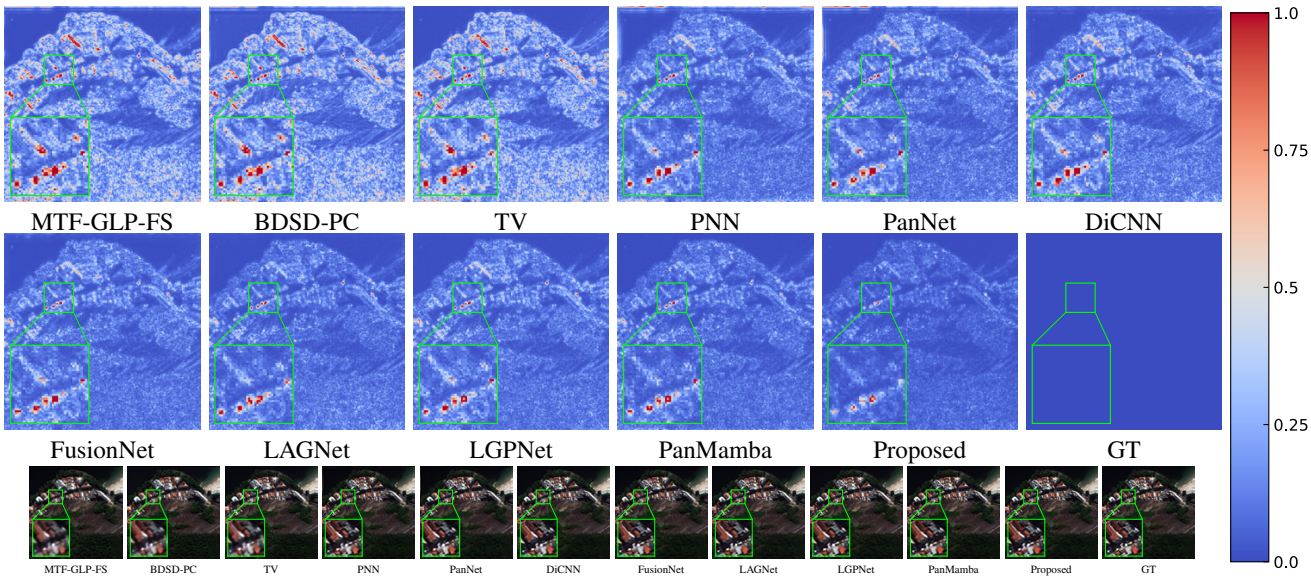

Figure 6. The residuals (Top) and visual results (bottom) of all compared approaches on the WV3 reduced-resolution dataset.

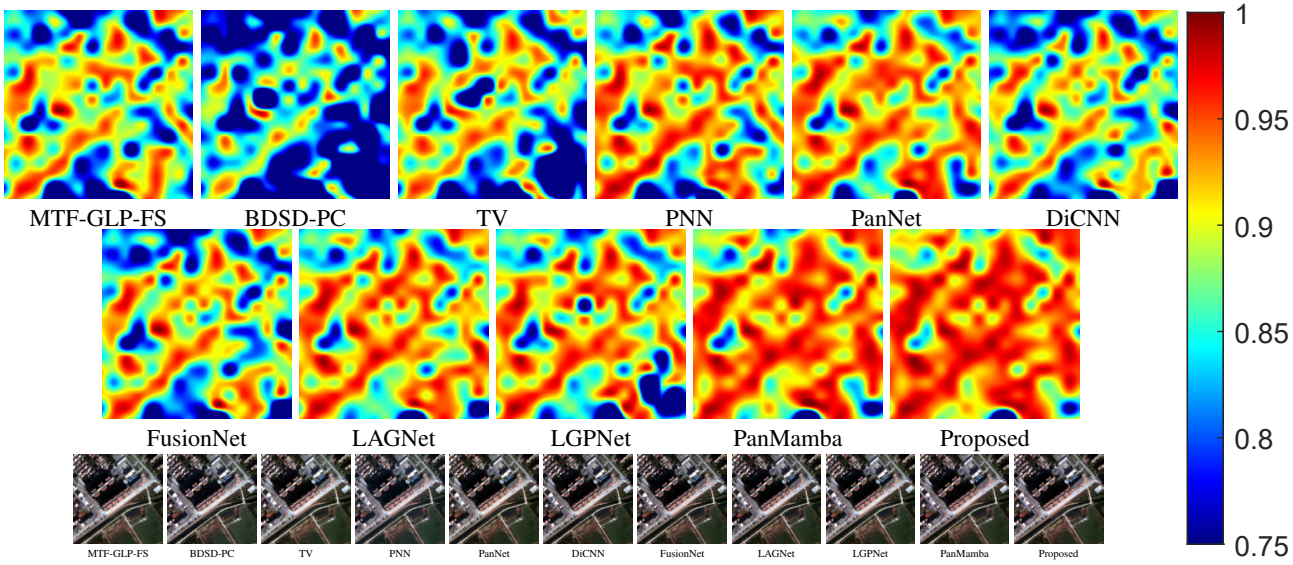

Figure 7. The HQNR maps (Top) and visual results (bottom) of all compared approaches on the GF2 full-resolution dataset.

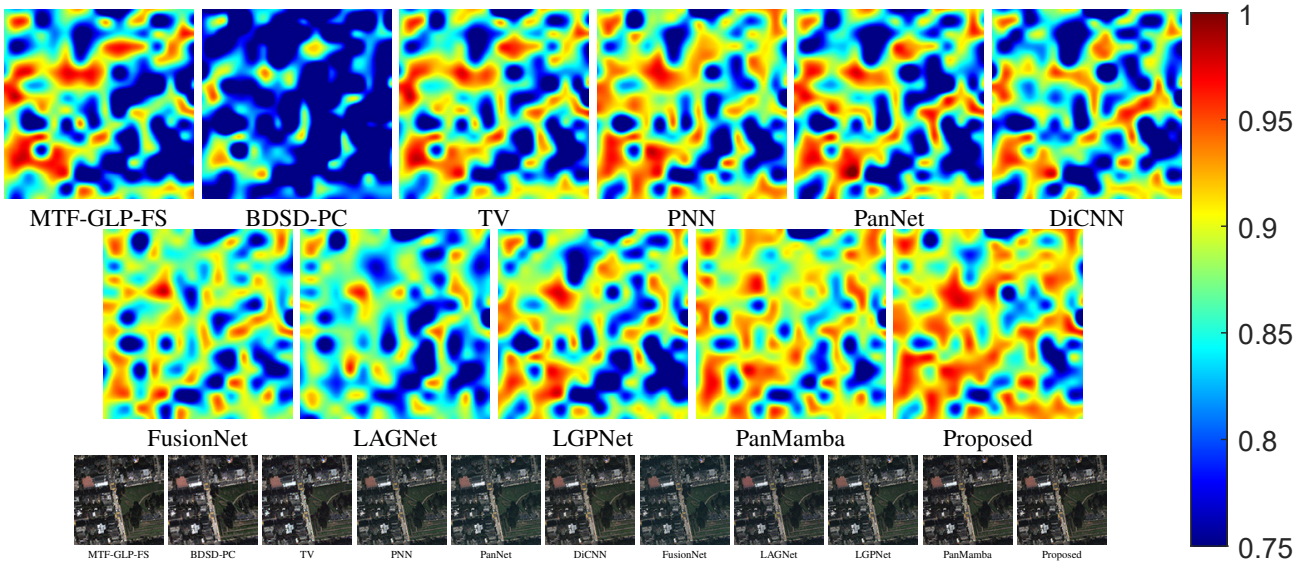

Figure 8. The HQNR maps (Top) and visual results (bottom) of all compared approaches on the QB full-resolution dataset.

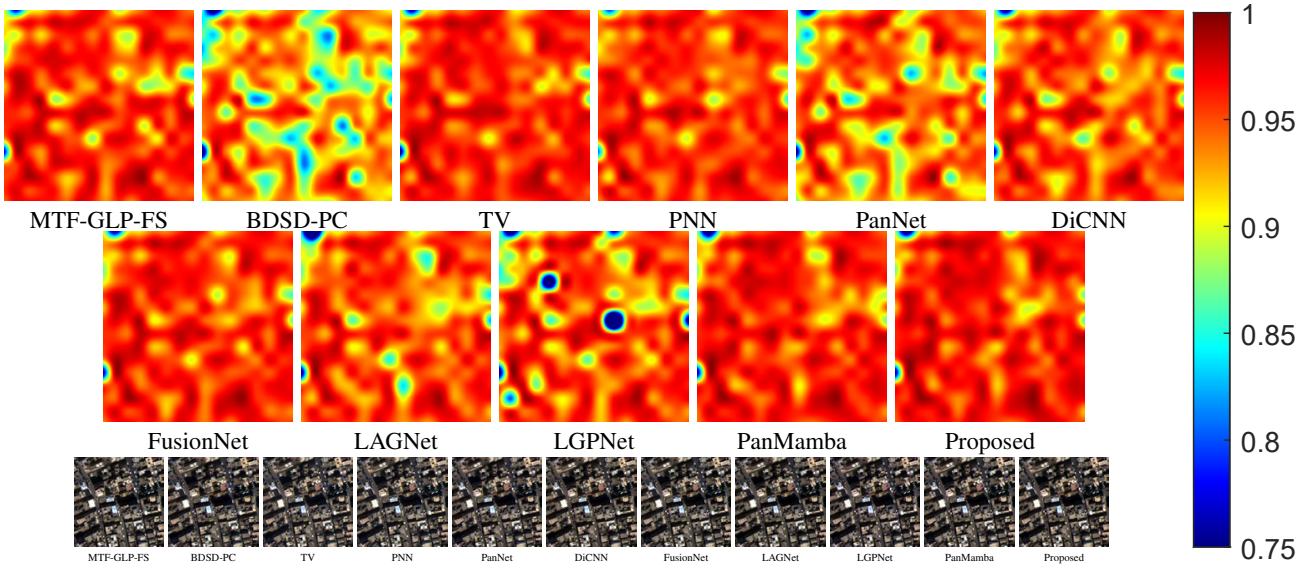

Figure 9. The HQNR maps (Top) and visual results (bottom) of all compared approaches on the WV3 full-resolution dataset.

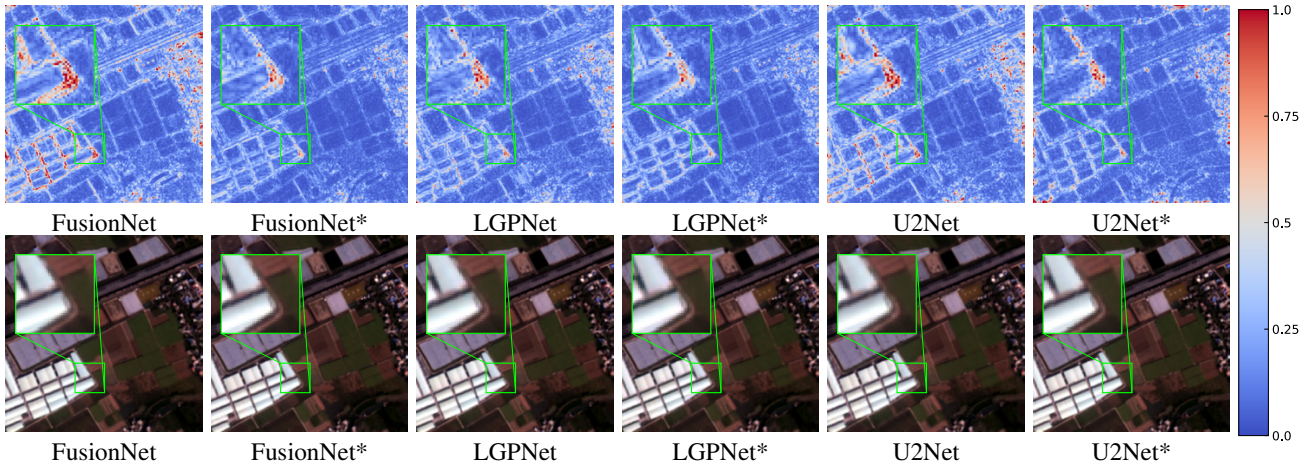

Figure 10. The residuals (Top) and visual results (Bottom) of all evaluated general methods on the GF2 reduced-resolution dataset.

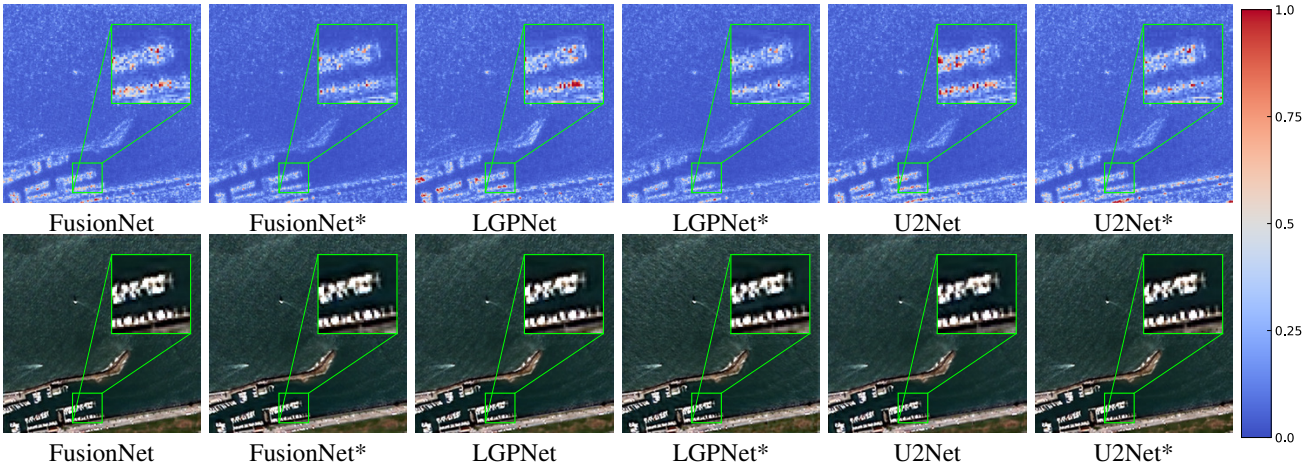

Figure 11. The residuals (Top) and visual results (Bottom) of all evaluated general methods on the QB reduced-resolution dataset.

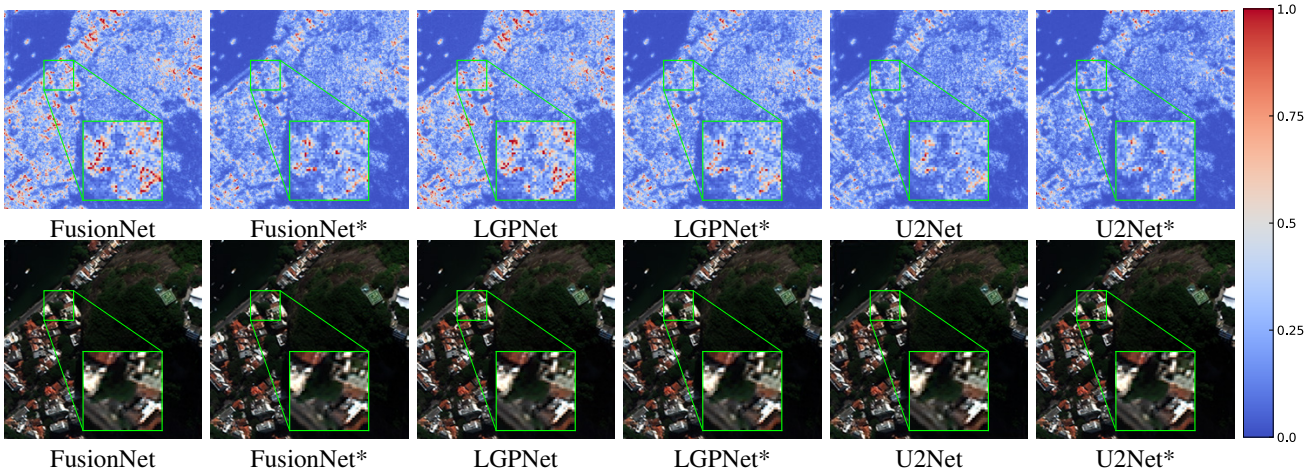

Figure 12. The residuals (Top) and visual results (Bottom) of all evaluated general methods on the WV3 reduced-resolution dataset.

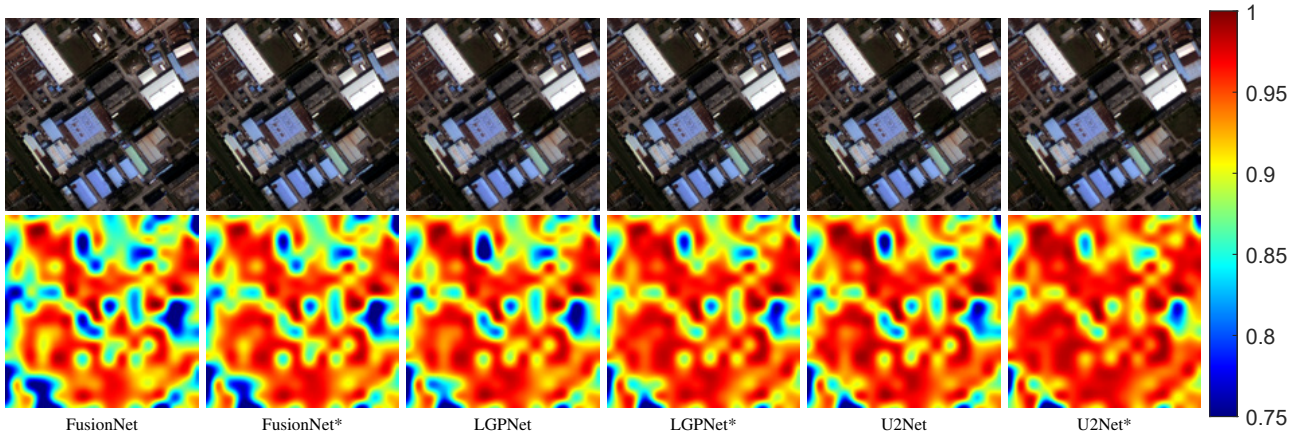

Figure 13. The visual results (Top) and HQNR maps (Bottom) of all evaluated general methods on the GF2 full-resolution dataset.

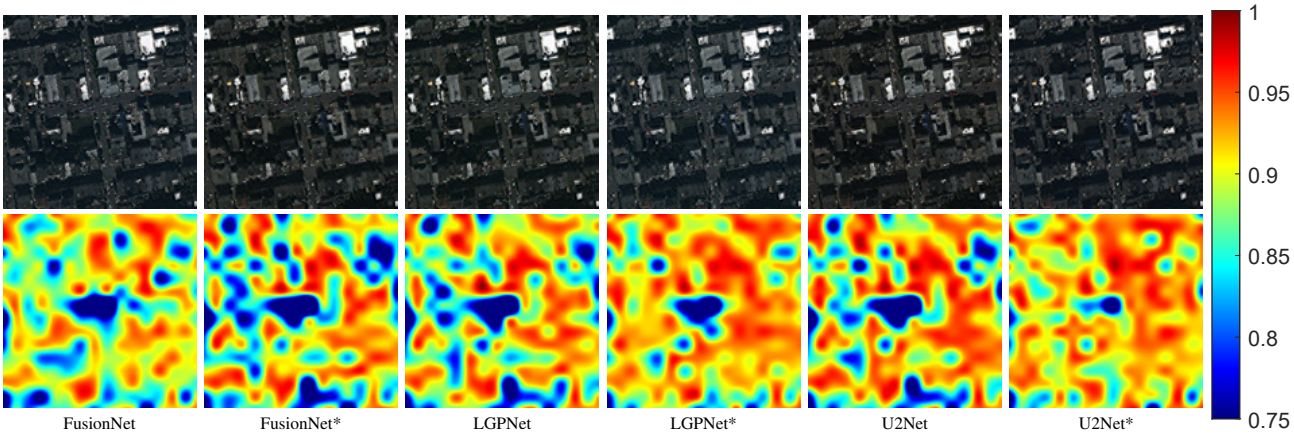

Figure 14. The visual results (Top) and HQNR maps (Bottom) of all evaluated general methods on the QB full-resolution dataset.