Cascadic Multireceptive Learning for Multispectral Pansharpening

Jun-Da Wang, Liang-Jian Deng, *Senior Member, IEEE*, Chen-Yu Zhao, Xiao Wu, Hong-Ming Chen, and Gemine Vivone, *Senior Member, IEEE*

***Abstract*— Pansharpening refers to the fusion of a panchromatic image with high spatial resolution (PAN) a multispectral image with low spatial resolution (LRMS) image with low spatial resolution to obtain a high spatial resolution multispectral (HRMS) image, which is beneficial to visual display and geographic research. Recently, many deep learning (DL) methods have been proposed to address the pansharpening problem, but still a few examples of DL-based techniques are designed from the perspective of a better receptive field while the scale of features greatly varies among different ground objects. In this article, we mainly focus on designing a cascadic multireceptive learning resblock (CML-resblock) relying on the residual network (ResNet) block, which can efficiently extract multiscale features from both the PAN and LRMS images. Moreover, we propose a novel multiplication network preserving a physical significance, which uses deep neural networks (DNNs) to learn the coefficients of the pixelwise restoration mapping and multiplies the upsampled LRMS image with the learned coefficients to get the HRMS image. The two parts mentioned above constitute our cascadic multireceptive learning network (CMLNet). Extensive experiments on both reduced-resolution and full-resolution images acquired by the WorldView-3 (WV-3), GaoFen-2 (GF-2), and QuickBird (QB) satellites show that the proposed approach outperforms state-of-the-art methods. Furthermore, additional experiments have been conducted to prove the generality of the CML-resblock and multiplication network. The code is available at: https://github.com/wajuda/CML.**

***Index Terms*— Cascadic multireceptive learning, deep convolutional neural networks (CNNs), image fusion, multiplication network, multispectral imaging, pansharpening, remote sensing.**

Manuscript received 11 September 2023; revised 25 October 2023; accepted 28 October 2023. Date of publication 3 November 2023; date of current version 20 November 2023. This work was supported in part by NSFC under Grant 12271083; in part by the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0501, Grant 2023NSFSC1341, and Grant 2022NSFSC1821; in part by the Key Projects of Applied Basic Research in Sichuan Province under Grant 2020YJ0216; and in part by the National Key Research and Development Program of China under Grant 2020YFA0714001. *(Corresponding author: Liang-Jian Deng.)*

Jun-Da Wang is with the Yingcai Honors College, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China (e-mail: 2020080601013@std.uestc.edu.cn).

Liang-Jian Deng and Xiao Wu are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China (e-mail: liangjian.deng@uestc.edu.cn; wxwsx1997@gmail.com).

Chen-Yu Zhao is with the School of Computer Science Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China (e-mail: chenyuzhaouestc@gmail.com).

Hong-Ming Chen is with the Key Laboratory of Oceanographic Big Data Mining and Application of Zhejiang Province, Zhejiang Ocean University, Zhoushan 316022, China, and also with the College of Electronics Information and Optical Engineering, Nankai University, Tianjin 300350, China (e-mail: 2022048@zjou.edu.cn).

Gemine Vivone is with the National Research Council—Institute of Methodologies for Environmental Analysis (CNR-IMAA), 85050 Tito Scalo, Italy, and also with National Biodiversity Future Center (NBFC), 90133 Palermo, Italy (e-mail: gemine.vivone@imaa.cnr.it).

Digital Object Identifier 10.1109/TGRS.2023.3329881

NOMENCLATURE

MS	LRMS image.
$\widetilde{\mathbf{MS}}$	LRMS image upsampled to PAN scale.
$\widehat{\mathbf{MS}}$	HRMS image.
P	PAN image.
RM	Coefficients of the restoration mapping.
GT	Ground-truth image.
RF	Receptive field size.

I. INTRODUCTION

HIGH spatial resolution multispectral (HRMS) images are widely used in many research fields since they can reflect changes in geographic information in a very accurate way. However, due to some physical constraints about the signal-to-noise ratio (SNR) of the acquired images by sensors onboard satellite platforms, high spatial and spectral resolutions are hardly achieved together by exploiting a single sensor. Hence, pansharpening, which stands for a panchromatic image with high spatial resolution (PAN) sharpening, is gaining attention in the literature. This technique aims to fuse a PAN image with a multispectral image with low spatial resolution (LRMS) image to get an HRMS image. Moreover, pansharpening has proven to be a powerful and effective image fusion methodology [3], [4], helpful in visual interpretation and as a preliminary step for further high-level image processing tasks. Multispectral pansharpening has also been extended to address similar tasks as hyperspectral pansharpening [5], [6] and multispectral and hyperspectral image fusion [7].

Over the past few decades, a large variety of pansharpening methods have been proposed. These approaches can be divided into four categories [3], [8], [9], i.e., component substitution (CS) methods, multiresolution analysis (MRA) methods, variational optimization-based (VO) methods, and deep learning (DL) methods. Our approach is based on convolutional neural networks (CNNs), belonging to the category of DL techniques. A brief introduction of some representative methods for each category is presented as follows.

The CS methods are based on the projection of the LRMS image into a transformed domain, where the LRMS component retaining most of the spatial information can be (partially or totally) substituted by the PAN image. Thanks to their simplicity, many pioneering algorithms have been developed belonging to this class [10], [11]. Some powerful instances are the band-dependent spatial detail (BDSD) [12] and its robust version (BDSD-PC) [13], the partial replacement adaptive CS (PRACS) [14], the Gram–Schmidt (GS) [15], and the Brovey

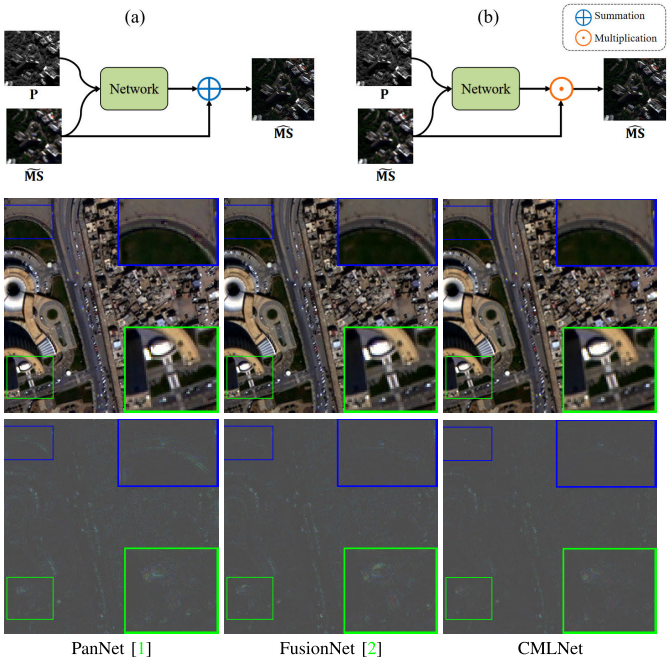


Fig. 1. First row: (a) overview of the pansharpening network architecture based on the additive injection model, i.e., the extracted spatial details from the PAN image (i.e., **P**) are injected into the upsampled LRMS image (i.e., $\widetilde{\mathbf{MS}}$) to get the fused product (i.e., $\widehat{\mathbf{MS}}$). (b) Our network architecture (i.e., CMLNet), which relies on the learning of the coefficients of the restoration mapping that will be multiplied by the upsampled LRMS image to get the HRMS image. Second and third rows: the results and corresponding absolute error maps (AEMs) of three deep CNNs, i.e., PanNet (SAM/ERGAS/Q8 = 3.00/1.95/0.96) [1], FusionNet (2.83/1.75/0.97) [2], and the proposed CMLNet (2.61/1.46/0.98). A better estimation of the HRMS image with fewer errors, as shown by the related AEM, is obtained by the CMLNet.

transform with haze correction [16]. Generally, CS techniques are usually characterized by a high fidelity in rendering spatial details in the fused product, while producing a significant spectral distortion.

The MRA-based methods are directly applied in the original LRMS spatial domain using the multiscale decomposition. More specifically, they extract spatial details from the PAN image by simple low-pass filters or exploit multiresolution approaches. Afterward, the details are injected into the interpolated LRMS image having the same size as the PAN image. Some methods belonging to this class are, for instance, the smoothing-filter-based intensity modulation (SFIM) [17], the additive wavelet luminance proportional (AWLP) [18], the generalized Laplacian pyramid (GLP) [19], the GLP with high-pass modulation injection model (GLP-HPM) [20], and the GLP with regression injection scheme (GLP-Reg) [21]. The MRA methods show a high spectral consistency but suffer from a spatial point of view, in contrast to the CS-based products [3].

The VO methods rely on a model that describes the relationship between the PAN, LRMS, and HRMS images [22]. The problem is to figure out how to set the variables so that the PAN and LRMS images can be used to estimate the HRMS image through a cost function with fidelity and regulation terms. These methods show an elegant mathematical formulation and have a good spatio-spectral preservation [3], with lower arithmetic speed. Examples in this category are Bayesian methods [23], sparse representation-based approaches (which represent the HRMS image as a sparse linear combination of dictionary elements) [24], [25], [26], variational techniques [27], and low-rank methods [28].

Recently, DL methods (in particular, CNN-based methods) are getting greater and greater attention, thanks to their excellent performance in the nonlinear mapping task and their powerful ability to extract features [29], [30], [31], [32], [33], [34]. Leveraging on a huge number of parameters (NoPs) and large-scale training datasets, the DL methods can perform better than many other approaches belonging to the above-mentioned three classes. Masi et al. [35] proposed the first CNN (named PNN) for pansharpening using a simple three-layer structure. Inspired by PNN, many researchers developed various structures that rely on CNNs, such as the residual module in residual network (ResNet) [36], which was widely used for pansharpening [1]. Moreover, Deng et al. [2] proposed the FusionNet under the guidance of both the CS and MRA methods. To improve the generality of DL methods, domain adaptation (DA) techniques have been drawn to solve problems related to cross-scene hyperspectral images. For example, Zhang et al. [37] used a graph structure to characterize topological relationships, and Zhang et al. [38] proposed a multilevel DA network to integrate DA with multisource data collaboration. Besides, the unsupervised learning strategy has been introduced for pansharpening [39], [40], [41]. Ma et al. [39] proposed a novel unsupervised framework for pansharpening based on a generative adversarial network, named as Pan-GAN. Luo et al. [40] designed a new loss function for unsupervised training containing spatial constraints and measuring a spectral consistency. Furthermore, Xiong et al. [41] adopted the high-frequency component of the corresponding PAN image as the weight to enhance the spatial details of residual block output features. However, a few examples of DL-based techniques have been proposed considering a key factor in deep neural networks (DNNs), the receptive field. We believe that the scale of features greatly varies among different ground objects from multiple sensors which calls for the multiscale feature extraction ability. Furthermore, most of the existing CNN-based frameworks for pansharpening are based on the additive injection model and tend to design deeper and more complicated networks while ignoring the physical interpretation. Since there are essential differences between the LRMS and PAN images in the spatial and spectral domains, we claim that the network framework should emphasize interpretability and physical sound.

To address the problems above, we propose a novel cascadic multireceptive learning network (CMLNet) for multispectral pansharpening. For the receptive field problem, we design a cascadic multireceptive learning resblock (CML-resblock) for extracting the features with multiple receptive fields (multireceptive). Besides, we present a novel pansharpening framework based on the multiplicative injection model, aiming to learn the value of the restoration mapping, which multiplies the upsampled LRMS image to obtain the HRMS image.

The main contributions of this work can be summarized as follows.

1) Inspired by the traditional multiplicative injection model for pansharpening, we design the novel multiplication network structure (see Section III-A) to learn the coefficients of the restoration mapping. Instead, some previously developed networks only learn the nonlinear mapping to separately extract spatial details from LRMS and PAN images, thereby losing the spectral information.
2) A CML-resblock (see Section III-B) is proposed to extract information from different scales in a step-by-step manner. Specifically, every pixel of the output is able to perceive multiscale information through a cascade-like connection strategy, which is an efficient and effective multireceptive learning process.

The rest of this article is organized as follows. In Section II, the motivations and the related works will be briefly introduced. The proposed network is presented in Section III. The experimental analysis is instead provided in Section IV. Finally, discussion and conclusions are drawn in Sections V and VI, respectively.

II. Related Works and Motivations

A. Notation

The main notation used in this article is presented in Nomenclature.

B. Pansharpening Methods

The pansharpening methods can be categorized into four classes [3]. Some most relevant techniques for our work, i.e., CS and CNN-based solutions, will be briefly introduced below.

1) CS: The traditional CS methods project $\widetilde{\mathbf{MS}}$ into a new domain, where the spatial structure, i.e., the intensity component ($\mathbf{I}_L$), is well-separated from the spectral information. Then, $\widetilde{\mathbf{MS}}$ can be restored by replacing $\mathbf{I}_L$ with the (histogram-matched) PAN image. Finally, $\widehat{\mathbf{MS}}$ is obtained by applying the inverse projection. The CS methods can generate HRMS images with outstanding visual performance and spatial misalignment robustness [8]. The general fusion equation for the CS-based methods is as follows:

$$\widehat{\mathbf{MS}}_k = \widetilde{\mathbf{MS}}_k + \mathbf{G}_k \cdot (\mathbf{P} - \mathbf{I}_L) \tag{1}$$

where k indicates the kth spectral band, $\mathbf{G}_k$ is the injection (gain) matrix for each $k = 1, \ldots, C$, C is the number of spectral bands (channels), and $\cdot$ denotes the elementwise matrix multiplication.

2) CNN-Based Approaches: The CNN-based solutions belong to the DL class for pansharpening. They have been widely applied in this field, thanks to their powerful fitting capabilities. A general framework for the CNN-based methods relies on learning spatial details to restore the HRMS product as shown in Fig. 1(a). Thus, the process can be described as follows:

$$\widehat{\mathbf{MS}} = \widetilde{\mathbf{MS}} + \mathcal{F}_\theta(\widetilde{\mathbf{MS}},\ \mathbf{P}) \tag{2}$$

where $\mathcal{F}_\theta$ is the nonlinear mapping function depending on the network parameter θ.

C. Receptive Field in Vision Tasks

The receptive field in DL is defined as the size of the region in the input that produces the feature [42]. It measures the relationship between an output feature (of any layer) and an input region. Small or large receptive fields correspond to small- or large-scale features, respectively. However, for a single receptive field, the convolution features mainly focus on the area of interest and ignore other potential information. Thus, obtaining multiscale features with multireceptive field kernels is necessary.

Recently, several strategies have realized the multiscale feature extraction. For example, U-Net [43] and feature pyramid networks (FPNs) [44] use a sequence of upsampling and downsampling layers to extract features with different scales. Moreover, inception [45] exploits multibranch convolutional layers with different kernel sizes gaining computational efficiency and low parameter count. Besides, efficient skip connections are applied by ResNet [36] and DenseNet [46] to mix multiscale features, which strongly enhance the performance of the model. Afterward, Res2Net [47] has been designed to enlarge the range of the receptive fields by splitting features and applying group convolutions in each layer (the interested readers can find more details in Section II-D). Although there are many multiscale feature works in vision tasks, a few of them focus on pansharpening. Hence, deeper insights are required for multiscale representation with multireceptive fields for this particular image fusion task.

D. Res2Net

Res2Net [47] is a multiscale module for CNNs, which improves performance for many vision tasks, e.g., classification, object detection, and class activation mapping. Instead of a group of 3×3 convolutions, Res2Net applies smaller groups of convolutions and connects different convolution groups in a hierarchical residual-like structure. Thus, the output represents different multiscale features in various channels. More specifically, as shown in Fig. 2(b), Res2Net splits first the feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ (where H and W are the height and width of $\mathbf{X}$, respectively) into m feature map subsets along channels, denoted by $\mathbf{x}_i \in \mathbb{R}^{H \times W \times (C/m)}$, and generates the corresponding output subsets, $\mathbf{y}_i$, where $i \in \{1, 2, \ldots, m\}$. It is worth to be noted that $\mathbf{y}_1$ is equal to $\mathbf{x}_1$, and each subset $\mathbf{x}_i$ ($i \in \{2, \ldots, m\}$) has a corresponding 3×3 convolution, denoted by $\mathbf{Conv}_i(\cdot)$. Specifically, $\mathbf{x}_i$ is added to the output of $\mathbf{Conv}_{i-1}(\cdot)$, and, then, fed into $\mathbf{Conv}_i(\cdot)$. The final result of Res2Net ($\mathbf{Y} \in \mathbb{R}^{H \times W \times C}$) can be written as follows:

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i, & i = 1 \\ \mathbf{Conv}_i(\mathbf{x}_i), & i = 2 \\ \mathbf{Conv}_i(\mathbf{x}_i + \mathbf{y}_{i-1}), & 2 < i \leq m. \end{cases} \tag{3}$$

Since each 3×3 convolution receives feature information from all the previous convolutional operations, the output of $\mathbf{Conv}_i(\cdot)$ has a larger receptive field than its input. Finally, Res2Net has a multiscale ability at a granular level and different subsets in output include different receptive fields [see Fig. 2(b)]. In addition, the receptive field size of Res2Net

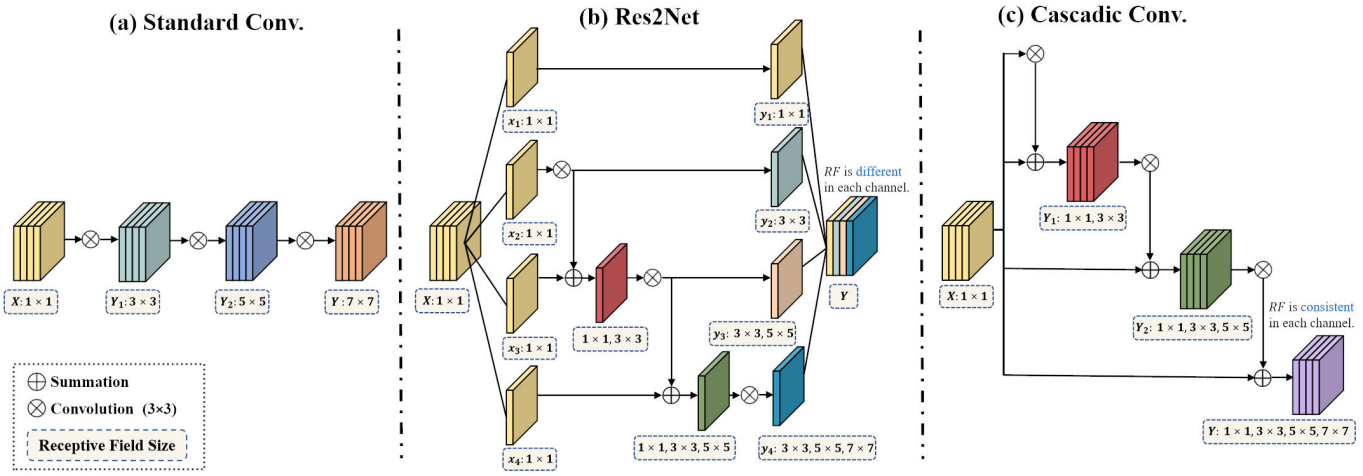


Fig. 2. Comparison of three different convolution operations. (a) Standard convolution. (b) Res2Net. (c) Proposed cascadic convolution. The features with different colors have different scales. For convenience, we omit the BatchNorm and rectified linear unit (ReLU) layers in this figure because they do not influence the receptive field. Note that these modules only replace the middle convolution layer of the bottleneck block before being plugged into the backbone of the networks.

can be described as follows:

$$\mathbf{RF}_i = \begin{cases} (2i-1)\times(2i-1), & 0 < i \le 2 \\ \mathbf{RF}_{i-1}, (2i-1)\times(2i-1), & 2 < i \le m \end{cases} \tag{4}$$

where $\mathbf{RF}_i$ is the receptive field size of $\mathbf{y}_i$.

E. Motivations

Despite the solutions discussed above, there is still room for improvement. Indeed, CNN-based techniques have been improved through the use of nonlinear mapping based on the additive injection model. However, the interpretability of the existing networks for pansharpening is limited by the increasingly sophisticated structures. Inspired by the traditional CS, we derive and propose a novel multiplication network with a simple structure and physical sound.

Moreover, even though Res2Net [47] has demonstrated its efficiency in extracting multiscale features, the performance is still far from optimality. Specifically, the split input feature subsets are processed in different multiscale extractions. Thus, the output contains various receptive fields in each subset, and several subsets only contain one scale information. Besides, without an in-depth fusion of the different receptive field information, the direct split and concatenation strategy restricts the ability of feature representation. These motivate us to design an equally efficient module that is more balanced and adequate for extracting multiscale features.

Considering the motivations above, the proposed network includes three main parts (shown in Fig. 3).

1) The novel multiplication network to learn a restoration map ($\mathbf{RM} \in \mathbb{R}^{H\times W\times C}$), which multiplies the preprocessed LRMS ($\widetilde{\mathbf{MS}}$) to obtain the HRMS image ($\widehat{\mathbf{MS}}$).
2) The cascadic multireceptive learning resblock (CML-resblock), which can extract multiscale features at pixel level with multireceptive fields.
3) The CMLNet consisting of the multiplication structure and CML-resblocks.

III. Proposed Network

A. Multiplication Network for Pansharpening

1) Proposed Multiplication Network: In the traditional injection scheme (1) of the CS framework, $\mathbf{I}_L$ in (1) is defined as follows:

$$\mathbf{I}_L = \sum_{i=1}^{C} \omega_i \widetilde{\mathbf{MS}}_i \tag{5}$$

where $\mathbf{w} = [w_1, \ldots, w_i, \ldots, w_C] \in \mathbb{R}^{1\times C}$ is the first row of the forward transformation matrix.

Unlike other networks that straightly replace the detail extraction process with nonlinear mapping of CNNs, we first define a space-varying injection gains $\mathbf{G}$ for each $k = 1, \ldots, C$ in (1) as follows:

$$\mathbf{G}_k = \frac{\widetilde{\mathbf{MS}}_k}{\mathbf{I}_L} = \frac{\widetilde{\mathbf{MS}}_k}{\sum_{i=1}^{C} \omega_i \widetilde{\mathbf{MS}}_i} \tag{6}$$

where the division is intended pixelwise. Then, we can rewrite (1) as follows:

$$\widehat{\mathbf{MS}}_k = \widetilde{\mathbf{MS}}_k + \frac{\widetilde{\mathbf{MS}}_k}{\mathbf{I}_L} \cdot (\mathbf{P} - \mathbf{I}_L) \tag{7}$$

$$= \widetilde{\mathbf{MS}}_k \cdot \frac{\mathbf{P}}{\mathbf{I}_L} \tag{8}$$

$$= \widetilde{\mathbf{MS}}_k \cdot \frac{\mathbf{P}}{\sum_{i=1}^{C} \omega_i \widetilde{\mathbf{MS}}_i}. \tag{9}$$

That is the classical high-pass modulation (HPM) (or multiplicative) injection scheme applied to CS pansharpening, which has recently been extensively studied by considering histogram-matching procedures [48], linear regression [49], or haze correction [16]. Moreover, (7) characterizes the fusion methods using the ratio of low-pass decompositions (ROLP), which has proven to be superior to the additive injection model (1) in preserving the visually important details of the component images and improving the quality of the fused products [20].

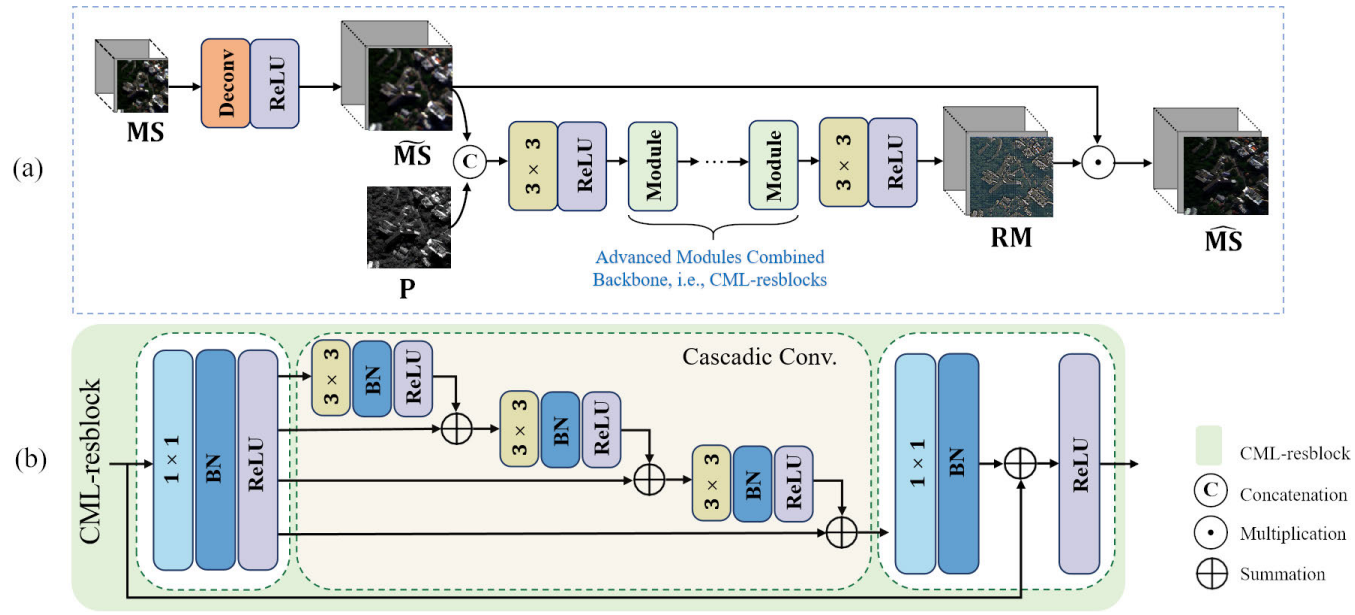


Fig. 3. Flowchart of (a) multiplication network and (b) CML-resblock, where the BN block indicates the batch normalization. The proposed CMLNet uses CML-resblocks as backbone blocks. The sizes of the data are as follows: $\mathbf{MS} \in \mathbb{R}^{h\times w\times C}$, $\widetilde{\mathbf{MS}} \in \mathbb{R}^{H\times W\times C}$, $\mathbf{P} \in \mathbb{R}^{H\times W}$, $\mathbf{RM} \in \mathbb{R}^{H\times W\times C}$, and $\widehat{\mathbf{MS}} \in \mathbb{R}^{H\times W\times C}$.

Thus, the pansharpening problem can be regarded as reconstructing $\widehat{\mathbf{MS}}$ from $\widetilde{\mathbf{MS}}$ with the restoration coefficients. We are able to construct the restoration coefficients by applying a properly chosen transformation from $\mathbf{P}$ and $\widetilde{\mathbf{MS}}$. Here, using the powerful nonlinear mapping ability, the coefficients ($\mathbf{RM}$) are estimated through CNNs and $\widetilde{\mathbf{MS}}$ is multiplied by $\mathbf{RM}$ to get $\widehat{\mathbf{MS}}$. In summary, our multiplication network can be expressed as follows:

$$\widehat{\mathbf{MS}} = \widetilde{\mathbf{MS}} \cdot \mathbf{RM} = \widetilde{\mathbf{MS}} \cdot \mathcal{F}_\theta(\widetilde{\mathbf{MS}},\ \mathbf{P}) \tag{10}$$

where $\mathcal{F}_\theta$ is a nonlinear mapping function depending on the network parameters θ generating $\mathbf{RM} \in \mathbb{R}^{H\times W\times C}$.

Theoretically, we can design any complex internal structure using advanced modules to get high performance. However, to keep the network structure as simple as possible and further prove its effectiveness, we simply stack several CML-resblocks to compose the backbone, which is a multireceptive learning module.

2) Difference With Existing Network: As shown in Fig. 1, the key differences between the proposed multiplication network and the other existing networks are as follows.

1) Other networks most use the additive injection model, while our network is designed based on the multiplicative injection model, which provides better results than the former in the literature.
2) Many CNN-based methods replace the linear mapping in the theoretical CS/MRA formulation with CNNs. Instead, the proposed multiplication network uses CNNs with a simple structure to learn the coefficients of the restoration mapping, which is a more effective and easier combination of the theoretical formulation and CNNs, thus getting both a physical significance and the relevant nonlinear mapping abilities of CNNs.

B. Cascadic Multireceptive Learning Resblock

1) Expression for Multiscale Receptive Field: We use first a novel expression to clearly show the multiscale receptive field of features. We can define the initial receptive field as 1×1 when the input feature is fed. When it goes through a convolutional layer with kernel size $k \times k$, the receptive field scale changes to $k \times k$. Furthermore, the output's receptive field scale increases step by step as the convolution progresses. The receptive field can be expressed as follows:

$$\mathbf{RF} = \left(\sum_{i=1}^{s} k_i - s + 1\right) \times \left(\sum_{i=1}^{s} k_i - s + 1\right) \tag{11}$$

where $\mathbf{RF}$ stands for the receptive field, s represents the number of the convolutional layers, and $k_i \times k_i$ is the kernel size of the ith convolutional layer.

It is worth noting that the skip connection adds two features with different receptive field sizes. According to the definition of the receptive field, the receptive field scale of the fused feature is dependent on the larger size because the smaller area is included in the larger area. However, if we only record the larger one, on one hand, it results in a multiscale learning process that cannot be distinct. On the other hand, it can be inaccurate because the two mixed features are obtained by different inputs. Taking the above statements into consideration, the receptive field of the fused feature can be expressed as follows:

$$\mathbf{RF}(\mathbf{z}) = (\mathbf{RF}(\mathbf{z}_1), \mathbf{RF}(\mathbf{z}_2)) \tag{12}$$

where $\mathbf{z}$, $\mathbf{z}_1$ and $\mathbf{z}_2$ represent the fused feature and the two connected features, respectively. These latter satisfies the relationship: $\mathbf{z} = \mathbf{z}_1 + \mathbf{z}_2$.

Furthermore, the convolutional operator only includes the summation and multiplication. When the fused feature enters

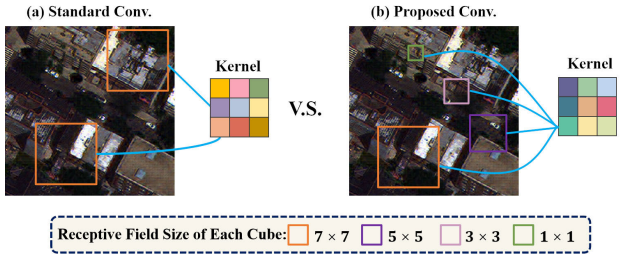


Fig. 4. Toy example to illustrate the use of the cascadic convolution. (a) Standard convolution operation, where all the pixels in the image/feature map are convolved with the same size. (b) Cascadic convolution operation with perceiving pixels at different scales in the image/feature map.

the next convolutional layer, the output is equal to the summation of the outputs of the two features coming into the same convolutional layer. Thus, the following relationship is valid:

$$\begin{aligned}\mathbf{RF}(\mathbf{Conv}(\mathbf{z})) &= \mathbf{RF}(\mathbf{Conv}(\mathbf{z_1}+\mathbf{z_2}))\\ &= (\mathbf{RF}(\mathbf{Conv}(\mathbf{z_1})), \mathbf{RF}(\mathbf{Conv}(\mathbf{z_2}))) \quad (13)\end{aligned}$$

where $\mathbf{Conv}(\cdot)$ is a convolutional layer.

Based on the above statements, we can make a thorough analysis of the changes in the receptive field. For instance, Fig. 2 provides a detailed breakdown of certain modules through the aforementioned statements. Moreover, this approach offers a comprehensive understanding of how the feature is affected by the changes in the receptive field and how it impacts the overall learning process (or functionality) of the system. By identifying these changes, we can address potential limitations in optimizing the network.

2) Cascadic Convolution: To alleviate the unbalanced multiscale feature extraction, we propose a new convolution method, i.e., cascadic convolution, which is a simple but effective module with a multireceptive field at pixel level.

In the cascadic convolution, we use a sequence of convolutional layers, and the output of each layer is added to the original input before entering into the next convolutional layer. Through this cascade-like connection strategy, the multiscale information is extracted step by step.

Fig. 2(c) shows how the cascadic convolution and the multireceptive learning process work. The input $\mathbf{X}$ directly enters a convolutional layer with kernel size 3×3 without any extra segmentation. The output's receptive field scale of $\mathbf{X}$ is 3×3 and it is added to $\mathbf{X}$ to generate $\mathbf{Y}_1$. The self-receptive field of $\mathbf{X}$ is 1×1. Thus, according to (12), $\mathbf{Y}_1$ receptive field is $(1 \times 1, 3 \times 3)$. Like a cascade, $\mathbf{Y}_1$ flows into the next level. Through another convolutional layer, the 1×1 receptive field has been modified in 3×3, and the 3×3 receptive field has been changed in 5×5. According to (13), the receptive field of the output of the second convolutional layer is $(3 \times 3, 5 \times 5)$. Similarly, $\mathbf{Y}_2$ receptive field scale is $(1 \times 1, 3 \times 3, 5 \times 5)$. Finally, the receptive field of the output $\mathbf{Y}$ is $(1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7)$. Each part of $\mathbf{Y}$ learns multiscale information from $\mathbf{X}$. Thus, the cascadic convolution is related to a pixelwise multireceptive learning process. Finally, to reduce the NoPs, the group convolution is used in the cascadic convolution without changing any key feature of this latter.

3) Proposed CML-Resblock: The idea of cascadic convolution partly stems from Res2Net. Thus, we choose the bottleneck block in ResNet as the framework for our module and use cascadic convolution to replace the middle convolutional layer of Res2Net.

In the original bottleneck block, three convolutional layers are stacked together, with kernel sizes of 1×1, 3×3, and 1×1, respectively. In the end, a parameter-free identity shortcut is used. In the proposed CML-resblock, the cascadic convolution takes the place of the second convolutional layer. Fig. 3 shows the complete structure of the CML-resblock, which can easily be plugged into other existing networks.

4) Differences From Standard Convolution: If we only stack convolutional layers as shown in Fig. 2(a), the receptive field scale increases step by step as the convolution progresses. According to (11), the output's receptive field scale becomes $(3 \times 3, 5 \times 5, 7 \times 7)$ through the three convolutional layers, respectively, with kernel size 3×3. This method is called standard convolution.

The proposed cascadic convolution has the following differences with respect to the standard convolution.

1) Although the maximum area of the input that can be seen by the output is the same as that of the cascadic convolution, standard convolution only obtains a single-scale receptive field limiting the performance of the modules, while CML-resblock is a multireceptive learning module, as shown in Fig. 4.
2) The output of each convolutional layer in the cascadic convolution is added to $\mathbf{X}$ before entering into the next layer. Namely, the deep-level feature extracted by the CML-resblock is always under the guidance of the input. Instead, in the standard convolution, the input acts only in the first convolutional layer.

5) Differences From Res2Net: The proposed cascadic convolution has some main differences with respect to the Res2Net module.

1) In the Res2Net module, for different subsets, $\mathbf{x}_i$, the extracted multiple scales are not the same. Moreover, the receptive fields vary with the channels of the output. Instead, in the cascadic convolution, each part of $\mathbf{X}$ gets the same and full multi-scale extraction and each part of $\mathbf{Y}$ perceives information from the same multiple scales of $\mathbf{X}$.
2) The strategy of the hierarchical connection directly adds different channels of the feature into Res2Net, which could reduce the diversity of the features, thus decreasing the feature representation ability. Instead, the cascadic convolution adds the corresponding positions of the features at different levels, thus retaining the diversity.
3) The group convolution used in the cascadic convolution does not change its critical properties. More specifically, although the filters are grouped, each part of $\mathbf{X}$ is subject to the same cascadic multireceptive operation, while different segments of $\mathbf{X}$ from Res2Net are subject to distinct operations.

C. Overall Structure of the Proposed Network

It is worth to be remarked that the scale of **MS** is different from that of **P** and **GT**. Thus, we need to upsample **MS** to the PAN scale, i.e., $\widetilde{\mathbf{MS}}$. To this aim, we use a deconvolution layer for the upsampling.

CS and MRA have a physical significance to extract detailed information to be injected into $\widetilde{\mathbf{MS}}$. Specifically, the CS-based methods rely on the spectral model ruling the projection of $\widetilde{\mathbf{MS}}$ into **P**. They subtract **P** with a linear combination of $\widetilde{\mathbf{MS}}$. While the MRA-based methods subtract **P** with its low-pass version $\mathbf{P}_L$. However, considering that our network just aims to learn the coefficients of the restoration mapping, we use as much information as possible and we just concatenate $\widetilde{\mathbf{MS}}$ and **P** as input for our network. This solution has two benefits. Indeed, we reduce the computational burden and we avoid the loss of spectral and spatial information.

At the beginning and end of the network, we, respectively, adopt a convolutional layer to change the number of channels. It is worth mentioning that the number of feature channels is the same in the backbone of the network. Afterward, we use a sequence of CML-resblocks to fully extract multiscale information from $\widetilde{\mathbf{MS}}$ and **P**. The multiscale extracting process of the CML-resblock is balanced and each part of the output perceives the same multiscale information from the input. Thus, we can obtain a multireceptive pixelwise coefficient, **RM**, to reconstruct $\widehat{\mathbf{MS}}$ in a better way. In our experiments, we empirically tuned the number of CML-resblocks and convolution kernel sizes according to the pansharpening literature and we set the number of filter groups in a proper way to have a similar NoPs with some recent and representative CNN-based methods. Finally, we multiply **RM** and $\widetilde{\mathbf{MS}}$ in a pixelwise manner to obtain $\widehat{\mathbf{MS}}$. The overall structure is shown in Fig. 3. Since the CML-resblocks represent the backbone of our network, this latter is called the CMLNet.

D. Loss Function

Since the main contributions of the proposed methods are the CML-resblock and multiplication structure, we empirically select the commonly used L1 loss function for usage, which mainly depicts the difference between groud-truth image (GT) and network output. Compared with another widely used L2 loss, this loss can generally preserve the sharp image edges and textures better so it is more widely used in the field of image processing. Specifically, the loss function is defined as follows:

$$\mathcal{L}oss = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{GT}^i - \widetilde{\mathbf{MS}}^i \cdot \mathcal{F}_\theta(\widetilde{\mathbf{MS}}^i,\ \mathbf{P}^i)\|_1 \qquad (14)$$

where $\|\cdot\|_1$ indicates the ℓ_1-norm, N is the number of training samples, and $\cdot^i$ is the ith training sample.

IV. Experiments and Results

In this section, to demonstrate the superiority of our CMLNet, we will compare it with some state-of-the-art pansharpening methods on the WorldView-3 (WV-3) dataset. Then, we will extend the performance assessment to two four-band datasets acquired by GaoFen-2 (GF-2) and QuickBird (QB). Finally, in the ablation studies, we will plug the CML-resblocks into other networks' backbones and replace the multiplication network with other network structures fixing the backbone to demonstrate the effectiveness of the proposed methods, respectively.

A. Datasets and Simulation

1) Datasets: We will consider several datasets captured by WV-3, GF-2, and QB. WV-3 acquires data in the visible near-infrared spectrum providing eight MS spectral bands (coastal, blue, green, yellow, red, red edge, near-infrared 1, and near-infrared 2) and a PAN band with a spatial resolution of 1.2 and 0.3 m, respectively, and a radiometric resolution of 11 bits. GF-2 and QB provide four MS spectral bands (red, green, blue, and near-infrared). Specifically, GF-2 acquires LRMS and PAN images with a spatial resolution of 4 and 1 m, and a radiometric resolution of 10 bits. Instead, QB captures LRMS and PAN images with a spatial resolution of 2.4 and 0.6 m, and a radiometric resolution of 11 bits. The above-mentioned datasets are publicly available.[1][2]

2) Simulation: It is worth noting that the datasets only include original PAN/LRMS image pairs. To get a reference (ground-truth) image, we need to follow Wald's protocol [50] to get **P**/**MS**/**GT** pairs for training. First, we filter the original images with the corresponding sensor's modulation transfer functions (MTFs), and then decimate the filtered images by a factor of 4. Afterward, we divide these simulated images into small patches.

For the WV-3 dataset, we simulated 12 580 **P**/**MS**/**GT** pairs with size 64 × 64, 16 × 16 × 8, and 64 × 64 × 8, respectively, and then split them into 70%/20%/10% for training (8806 examples)/validation (2516)/testing (1258).

Besides, other two WV-3 test cases, i.e., the Rio dataset and the Tripoli dataset, are used to test the generality of our method. They consist of **P**/**MS**/**GT** pairs with size 256 × 256, 64 × 64 × 8, and 256 × 256 × 8, respectively.

For the QB case, we downloaded a large dataset (4906 × 4906 × 4) acquired over the city of Indianapolis (USA) cutting it into two parts. The left part (4906 × 3906 × 4) is used to simulate 20 685 training samples (64 × 64 × 4), and the right part (4906 × 1000 × 4) is used to simulate 48 testing data (256 × 256 × 4).

For the GF-2 case, we downloaded a large dataset (6907 × 7300 × 4) captured over the city of Guangzhou (China) to simulate 19 809 training examples (64 × 64 × 4) and 20 testing examples (256 × 256 × 4).

It is worth to be noted that each pixel value of the input images is divided by 2047, for WV-3 and QB, and by 1023, for GF-2, to get the range [0, 1]. Except for this step, no further preprocessing (e.g., random augmentation) is performed.

B. Training Platform and Implementation Details

The proposed network is coded with Python 3.8.0 and PyTorch 1.7.0 and trained with an NVIDIA graphics processing unit (GPU) GeForce GTX 3080. We use the Adam optimizer, in which the weight decay is set as 1×10^{-8}.

[1]https://resources.maxar.com for the WV-3 and QB datasets.

[2]https://liangjiandeng.github.io/PanCollection.html for the GF-2 dataset.

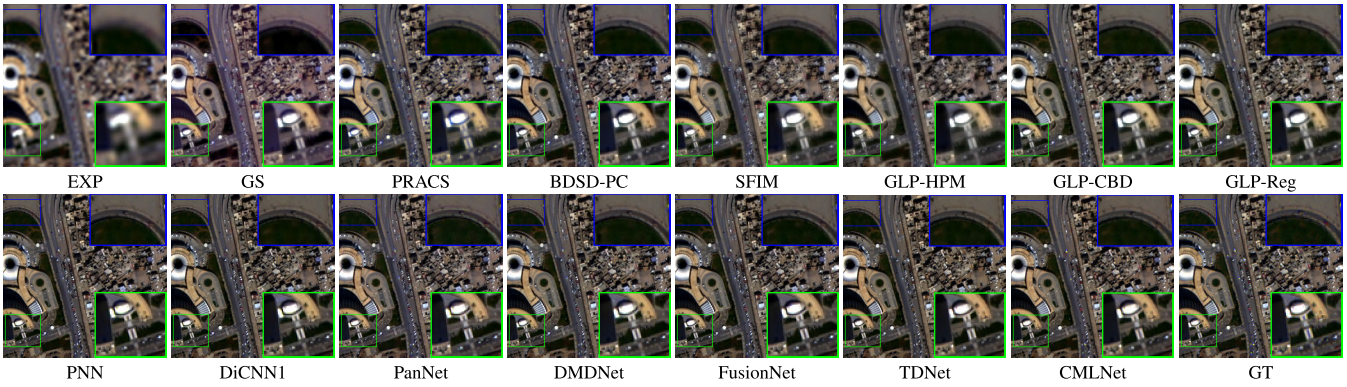


Fig. 5. Visual comparisons of all the approaches on the reduced-resolution Rio dataset (sensor: WV-3).

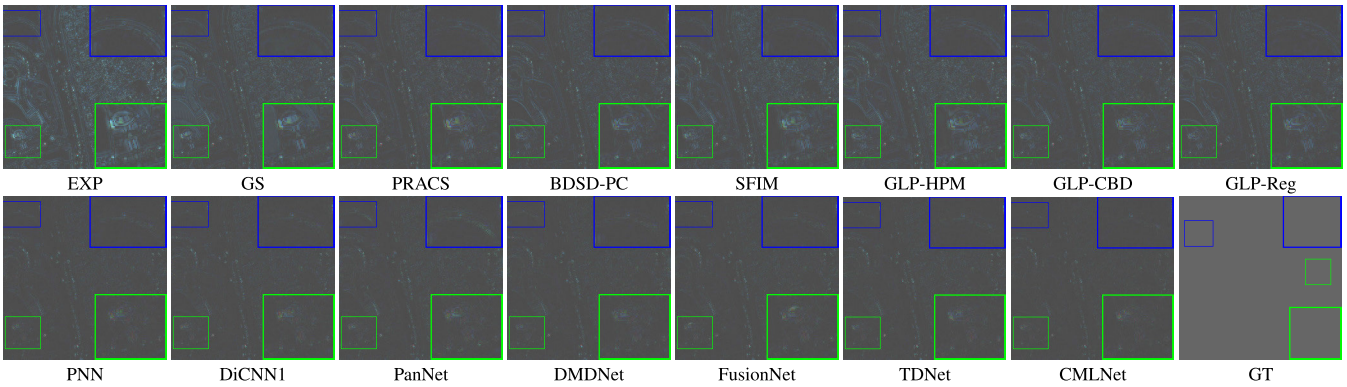


Fig. 6. Corresponding AEMs using the GT image on the reduced-resolution Rio dataset (sensor: WV-3). For better visualization, we doubled the intensities of the AEMs and added 0.3.

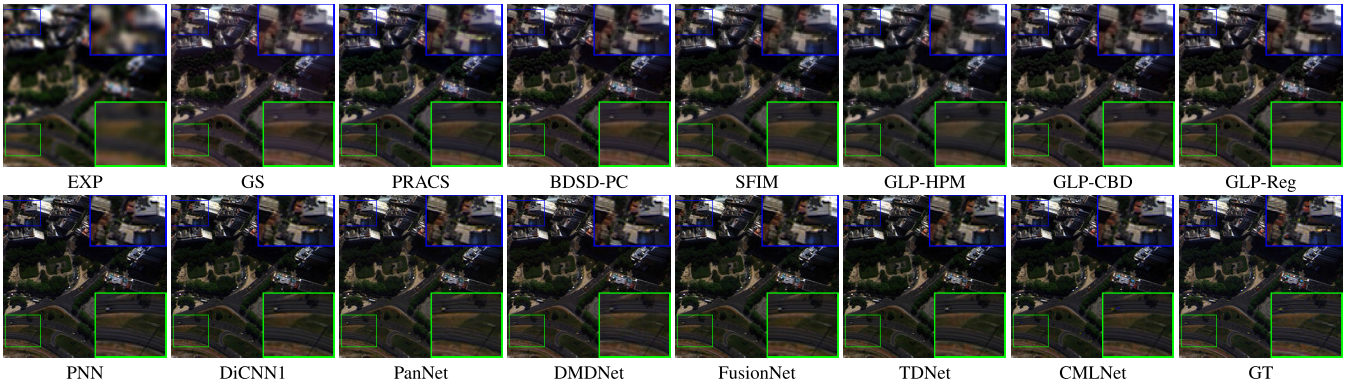


Fig. 7. Visual comparisons of all the approaches on the reduced-resolution Tripoli dataset (sensor: WV-3).

To achieve better performance, we set the initial learning rate to 0.0015, and then decrease it by one quarter every 200 epochs.

C. Parameters' Tuning

In our CMLNet, we chose the same parameter tuning as in [2]. Moreover, we simply stack four CML-resblocks in the backbone network and divide the filters in a convolutional layer of cascadic convolutions into 18 groups of four channels for each group. It is worth noting that this selection of parameters could not be optimal, but it turns out to be similar in the NoPs with respect to the compared approaches, thus demonstrating (in a fair way) the effectiveness of our method with respect to the benchmark.

D. Benchmark

Several state-of-the-art methods belonging to different pan-sharpening classes are used.

1) *EXP:* MS image interpolated by a polynomial kernel with 23 coefficients [19].
2) *CS Methods:*
   a) *GS:* GS sharpening approach [15].
   b) *PRACS:* PRACS approach [14].
   c) *BDSD-PC:* Robust BDSD approach [13].
3) *MRA Methods:*
   a) *SFIM:* SFIM approach [17].
   b) *GLP-HPM:* The GLP with MTF-matched filter [51] and multiplicative injection model [20].

TABLE I

AVERAGE METRICS FOR ALL THE COMPARED DL-BASED APPROACHES ON 1258 REDUCED-RESOLUTION SAMPLES FOR WV-3. (BOLD: BEST; UNDERLINE: SECOND BEST)

Method	*SAM*(± std)↓	*ERGAS*(± std)↓	*Q8*(± std)↑	*SCC*(± std)↑
PNN [35]	4.401 ± 1.329	3.228 ± 1.004	0.888 ± 0.112	0.921 ± 0.046
DiCNN1 [53]	3.980 ± 1.318	2.736 ± 1.015	0.909 ± 0.112	0.952 ± 0.047
PanNet [1]	4.092 ± 1.273	2.952 ± 0.977	0.894 ± 0.117	0.949 ± 0.046
DMDNet [54]	3.971 ± 1.248	2.857 ± 0.966	0.900 ± 0.114	0.952 ± 0.044
FusionNet [2]	3.743 ± 1.225	2.567 ± 0.944	0.913 ± 0.112	0.958 ± 0.045
TDNet [55]	3.503 ± 1.241	2.443± 0.958	0.921 ± 0.111	0.962 ± 0.044
CMLNet	**3.428 ± 1.173**	**2.381 ± 0.910**	**0.922 ± 0.107**	**0.965 ± 0.040**
Ideal value	**0**	**0**	**1**	**1**

c) *GLP-CBD:* The GLP with MTF-matched filter [51] and regression-based injection model [19], [52].

d) *GLP-Reg:* The GLP with MTF-matched filter [51] and full-scale regression injection model [21].

4) *DL-Based Methods:*

a) *PNN:* Pansharpening via CNNs [35].

b) *PanNet:* CNNs for residual learning on the high-frequency domain for pansharpening [1].

c) *DiCNN1:* Detail injection-based CNN [53].

d) *DMDNet:* Deep multiscale detail CNNs for pansharpening [54].

e) *FusionNet:* Deep CNN inspired by the traditional CS and MRA methods [2].

f) *TDNet:* Triple–double CNN motivated by the traditional MRA methods [55].

E. Reduced-Resolution Assessment

The reduced-resolution assessment measures the similarity between the fused image, $\widehat{\mathbf{MS}}$, and the related **GT**. The quality metrics used to assess the similarity are: the spectral angle mapper (SAM) [56], the dimensionless global error in synthesis (ERGAS) [57], the spatial correlation coefficient (SCC) [58], and the $Q2^n$ index [59] (Q4 and Q8 for four and eight bands, respectively). The ideal values are 0 for SAM and ERGAS, and 1 for SCC and $Q2^n$.

1) Experiments on the WV-3 Dataset: In Table I, we report the average quantitative results and the standard deviations of all the metrics for different methods on the testing dataset. In terms of the numerical analysis, the proposed CMLNet obtains the best average quantitative performance for all the quality indexes. Furthermore, the standard deviation (std) of all the metrics is very small, which also demonstrates the robustness of the proposed network. Compared with PanNet and FusionNet, whose input images are processed before inputting them into the network, our approach just feeds the network with the concatenation of **P** and $\widetilde{\mathbf{MS}}$. Moreover, to improve the performance, many CNN-based methods design special modules and network structures for learning the nonlinear relationship. For instance, TDNet relies on a complicated double-branch structure and several modules to extract information. Instead, we just propose a general and efficient multiscale extraction module (CML-resblock), simply stacking them into the network backbone.

2) Test on Two Different WV-3 Cases: A further test is introduced in this section by assessing the performance on two new cases from the Rio and Tripoli datasets without any adjustment or extra training. Table II shows the results obtained by all the methods. The quantitative metrics also demonstrate the superiority of CMLNet. Besides, Figs. 5–8 show the visual comparisons among all the compared pansharpening approaches. The visual analysis in Figs. 5 and 7 corroborates the numeric results. From the specific areas framed by blue and green boxes, it is clear to see blur effects generated by the classic CS and MRA methods. The CNN-based methods have a better visual effect than the traditional approaches, but it is not so easy to distinguish their performance due to the 8-bit RGB representation of the 11-bit multispectral fused data. Thus, we leverage on the calculation of the AEMs in Figs. 6 and 8. It is easy to see that there are fewer bright spots in the error maps of our results and they are closer to the gray value, indicating that the CMLNet products are the most similar to the **GT** data. Furthermore, we present in Fig. 9 the corresponding **RM** obtained by the proposed CMLNet as grayscale maps, one for each band. It can be observed that the value of each pixel of **RM** is around 1, which meets our initial assumption to see it as coefficients of restoration mapping. According to Fig. 9, **RM** contains a lot of spatial details, being involved in the reconstruction process.

TABLE II

QUALITY METRICS FOR ALL THE COMPARED APPROACHES ON THE REDUCED-RESOLUTION RIO AND TRIPOLI DATASETS. (BOLD: BEST; UNDERLINE: SECOND BEST)

	SAM↓	*ERGAS*↓	*Q4*↑	*SCC*↑
(a) Rio dataset				
EXP [19]	4.2030	5.5976	0.6927	0.6156
GS [15]	4.0614	3.8956	0.8666	0.8979
PRACS [14]	4.0260	3.2501	0.9062	0.8972
BDSD-PC [13]	3.8065	2.8494	0.9363	0.9061
SFIM [17]	3.9132	3.5630	0.8859	0.8880
GLP-HPM [20]	4.1349	3.4917	0.8935	0.8817
GLP-CBD [52]	3.7068	2.7732	0.9350	0.9092
GLP-Reg [21]	3.6871	2.7760	0.9345	0.9095
PNN [35]	3.3728	2.3082	0.9488	0.9409
DiCNN [53]	3.0248	1.9119	0.9686	0.9627
PanNet [1]	3.0054	1.9506	0.9651	0.9640
DMDNet [54]	2.9355	1.8119	0.9691	0.9699
FusionNet [2]	2.8338	1.7510	0.9728	0.9714
TDNet [55]	2.7373	1.6733	0.9764	0.9756
CMLNet	**2.6176**	**1.4605**	**0.9803**	**0.9820**
(b) Tripoli dataset				
EXP [19]	6.7883	8.5719	0.7235	0.5129
GS [15]	7.1416	7.3237	0.7879	0.7251
PRACS [14]	6.6680	7.0012	0.8266	0.7253
BDSD-PC [13]	6.4985	6.7186	0.8475	0.7313
SFIM [17]	6.3486	6.8407	0.8343	0.7341
GLP-HPM [20]	6.8196	6.8881	0.8393	0.7350
GLP-CBD [52]	6.4178	6.5443	0.8503	0.0.7392
GLP-Reg [21]	6.4100	6.5463	0.8548	0.7394
PNN [35]	5.0778	3.9614	0.9214	0.9242
DiCNN [53]	4.7552	3.4978	0.9444	0.9482
PanNet [1]	4.6079	3.4227	0.9395	0.9516
DMDNet [54]	4.4282	3.1972	0.9458	0.9613
FusionNet [2]	4.2764	3.0568	0.9522	0.9646
TDNet [55]	4.1277	3.0471	0.9542	0.9658
CMLNet	**3.7900**	**2.7091**	**0.9625**	**0.9744**
Ideal value	**0**	**0**	**1**	**1**

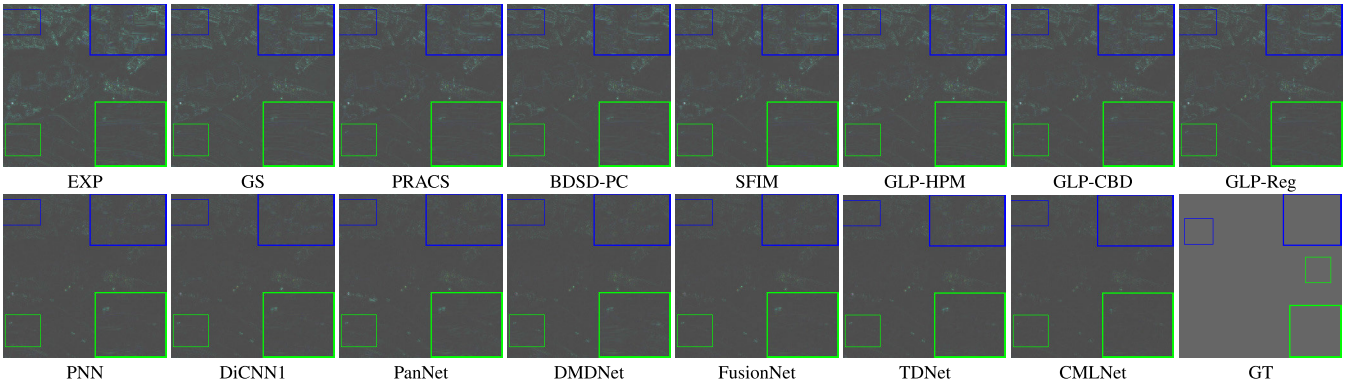


Fig. 8. Corresponding AEMs using the GT image on the reduced-resolution Tripoli dataset (sensor: WV-3). For better visualization, we doubled the intensities of the AEMs and added 0.3.

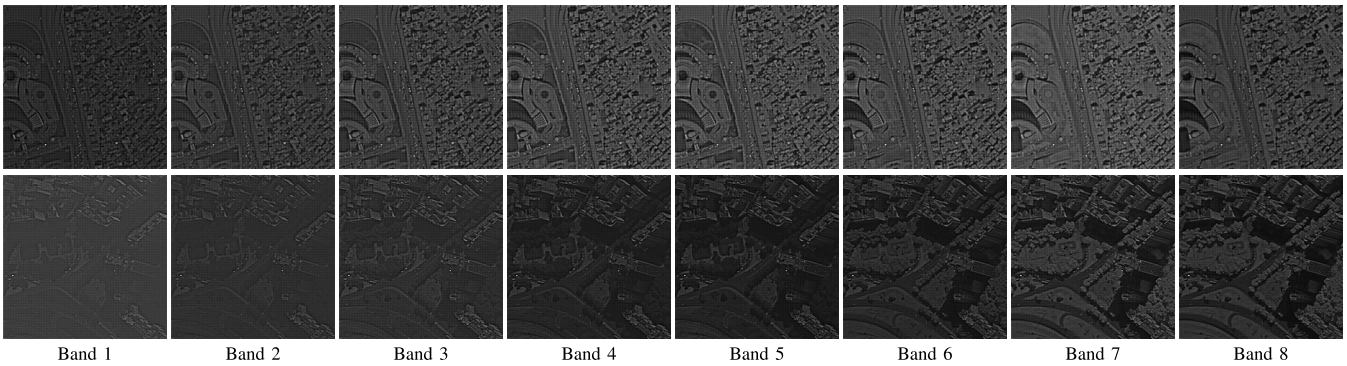


Fig. 9. **RM** learned by the proposed CMLNet on the reduced-resolution Rio and Tripoli datasets (sensor: WV-3).

TABLE III

COMPARISON OF THE TRAINING TIMES, THE TESTING TIMES, THE NOPS, AND THE GFLOPS FOR ALL THE DL-BASED APPROACHES. (THE TRAINING TIMES UNIT IS HOURS: MINUTES, AND THE TESTING TIMES UNIT IS SECONDS)

	PNN	DiCNN1	PanNet	BDPN	DMDNet	FusionNet	TDNet	CMLNet
Training time	25: 15	7: 06	4: 32	46:19	5: 27	2: 21	6: 30	3: 25
Testing time	0.0778	0.0799	0.0811	0.0912	0.0852	0.0812	0.0861	0.0827
NoPs	10.4×10^4	4.6×10^4	8.3×10^4	148.4×10^4	10.0×10^4	7.8×10^4	54.5×10^4	8.8×10^4
GFLOPs	0.29	0.19	0.23	3.8	0.52	0.32	1.38	0.35

3) Computational Analysis: Table III reports the training times, the testing times, the NoPs, and the giga floating point operations per second (GFLOPs) for all the compared CNN-based methods using the WV3 dataset. It is worth to be remarked that the training iterations are determined when obtaining the optimal testing quantitative results. Furthermore, the average testing times are calculated on the same GPU. As mentioned above, the proposed CMLNet has a similar structure with respect to PanNet and FusionNet. Thus, they are also similar in terms of testing times, NoPs, and GFLOPs. More specifically, we used CML-resblocks in the proposed method containing cascadic convolutions, resulting in a slight increase in computational complexity. Moreover, TDNet gets a relatively large amount of parameters, mainly due to the triple–double structure and the multi-scale convolutional block (MSCB) for multiscale convolution. According to Table III and the related quantitative experimental results, it is clear that the proposed CMLNet gets a significant increase in performance without a related computational complexity growing.

F. Full-Resolution Assessment

The goal of pansharpening is to obtain an HRMS image for real applications, and thus, an assessment at full resolution is of crucial importance to state the superiority of the proposed approach. Since there is no GT at full resolution, we exploit the quality with no reference (QNR) index [60], consisting in a spectral distortion (D_λ) and a spatial distortion (D_s) indexes, as quality metric. The QNR has an ideal value equal to 1 obtained when both D_λ and D_s are equal to 0.

We exploit ten original PAN/LRMS image pairs of the full-resolution WV-3 dataset, named Tripoli-OS dataset. Table IV shows that our CMLNet obtains the best average results and standard deviations on the three metrics. Moreover, Fig. 10 shows the visual comparison for the Tripoli-OS dataset. We can easily see that there are some spatial distortions and blur effects inside the close-ups for the compared approaches except for the proposed one. To further investigate the performance at full resolution, Fig. 11 shows some full-resolution HRMS images obtained by fusing WV3 data with the proposed approach. Fig. 11 demonstrates that our method can get a high spatial fidelity retaining PAN spatial details.

G. Experiments on the GF-2 and QB Datasets

In this section, we assess the performance on two four-band datasets acquired by GF-2 and QB, respectively. Despite **MS**

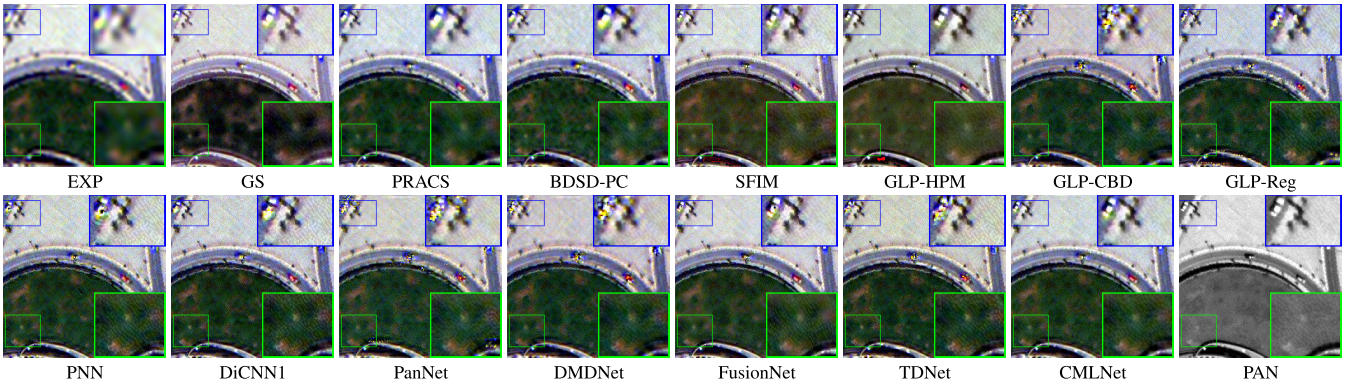


Fig. 10. Visual comparison among the approaches on the full-resolution Tripoli-OS dataset (sensor: WV-3).

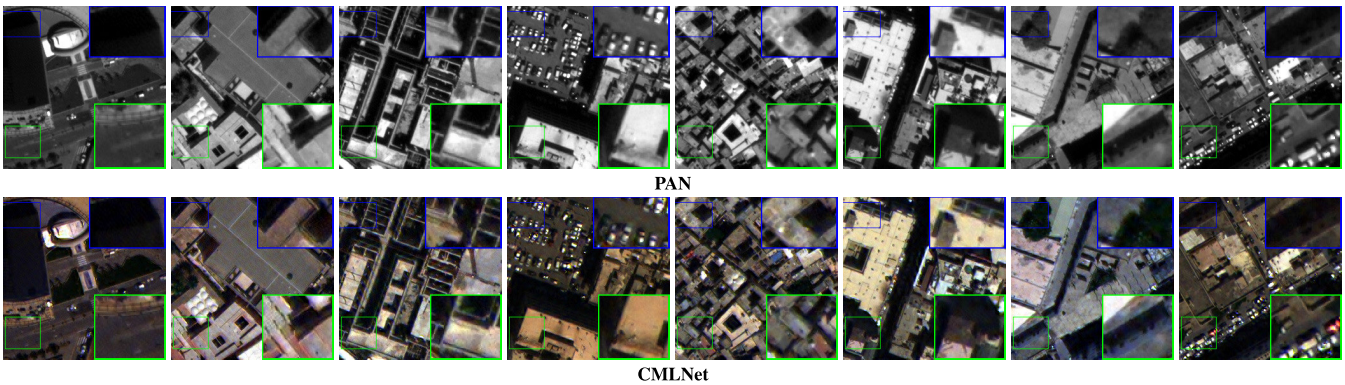


Fig. 11. Visual performance of the proposed CMLNet on the full-resolution Tripoli-OS dataset (sensor: WV-3). First row: the PAN images. Second row: the corresponding fused HRMS images.

TABLE IV

AVERAGE VALUES OF THE QNR, D_λ, AND D_s INDEXES WITH THE RELATED STANDARD DEVIATIONS (STDS) FOR THE TRIPOLI-OS DATASET. (BOLD: BEST; UNDERLINE: SECOND BEST)

Method	*QNR*(± std)↑	D_λ(± std)↓	D_s(± std)↓
EXP [19]	0.9092 ± 0.0267	0.0289 ± 0.0195	0.0907 ± 0.0267
GS [15]	0.8673 ± 0.0583	0.0298 ± 0.0206	0.1066 ± 0.0441
PRACS [14]	0.8819 ± 0.0615	0.0276 ± 0.0189	0.0937 ± 0.0470
BDSD [12]	0.9191 ± 0.0537	0.0209 ± 0.0093	0.0615 ± 0.0470
BDSD-PC [13]	0.8786 ± 0.0632	0.0317 ± 0.0197	0.0932 ± 0.0500
SFIM [17]	0.9037 ± 0.0454	0.0402 ± 0.0242	0.0588 ± 0.0247
GLP-HPM [20]	0.8284 ± 0.0704	0.0822 ± 0.0333	0.0987 ± 0.0464
GLP-CBD [52]	0.8768 ± 0.0619	0.0491 ± 0.0252	0.0787 ± 0.0431
GLP-Reg [21]	0.8789 ± 0.0589	0.0480 ± 0.0240	0.0775 ± 0.0411
PNN [35]	0.9348 ± 0.0277	0.0291 ± 0.018	0.0372 ± 0.0121
DiCNN1 [53]	0.9264 ± 0.0283	0.0254 ± 0.0147	0.0495 ± 0.0169
PanNet [1]	0.9450 ± 0.0172	0.0281 ± 0.0109	0.0276 ± 0.0075
DMDNet [54]	0.9499 ± 0.0153	0.0249 ± 0.0116	0.0257 ± 0.0062
FusionNet [2]	0.9405 ± 0.0658	0.0310 ± 0.0223	0.0426 ± 0.0135
TDNet [55]	0.9384 ± 0.0141	0.0258 ± 0.0111	0.0366 ± 0.0098
CMLNet	**0.9567 ± 0.0112**	**0.0214 ± 0.0069**	**0.0222 ± 0.0057**
Ideal value	**1**	**0**	**0**

TABLE V

AVERAGE RESULTS OF THE COMPARED APPROACHES FOR 20 GF-2 TESTING SAMPLES AND 48 QB TESTING SAMPLES. (BOLD: BEST; UNDERLINE: SECOND BEST)

	SAM(± std)↓	*ERGAS*(± std)↓	*Q4*(± std)↑	*SCC*(± std)↑
Guangzhou datasets (GaoFen-2)				
EXP [19]	1.845 ± 0.357	2.399 ± 0.478	0.791 ± 0.046	0.870 ± 0.031
PNN [35]	1.049 ± 0.219	1.059 ± 0.227	0.959 ± 0.009	0.977 ± 0.005
DiCNN1 [53]	1.053 ± 0.223	1.081 ± 0.244	0.958 ± 0.009	0.977 ± 0.005
PanNet [1]	0.998 ± 0.206	0.922 ± 0.185	0.966 ± 0.010	**0.982 ± 0.003**
FusionNet [2]	0.974 ± 0.205	0.989 ± 0.213	0.962 ± 0.009	0.980 ± 0.004
TDNet [55]	0.941 ± 0.172	0.892 ± 0.172	0.967 ± 0.013	0.975 ± 0.005
CMLNet	**0.905 ± 0.170**	**0.852 ± 0.166**	**0.971 ± 0.010**	0.966 ± 0.009
Indianapolis datasets (QuickBird)				
EXP [19]	8.156± 1.9571	11.567 ± 2.189	0.572 ± 0.106	0.524 ± 0.022
PNN [35]	5.799 ± 0.947	5.571 ± 0.458	0.857 ± 0.148	0.902 ± 0.048
DiCNN1 [53]	5.307 ± 0.995	5.231 ± 0.541	0.882 ± 0.143	0.922 ± 0.050
PanNet [1]	5.314 ± 1.017	5.162 ± 0.681	0.883 ± 0.139	0.929 ± 0.058
DMDNet [54]	5.119 ± 0.939	4.737 ± 0.648	0.890 ± 0.146	0.935 ± 0.065
FusionNet [2]	4.540 ± 0.778	4.050 ± 0.266	0.910 ± 0.136	0.954 ± 0.045
TDNet [55]	4.504 ± 0.802	3.979 ± 0.232	0.912 ± 0.145	0.955 ± 0.052
CMLNet	**4.486 ± 0.758**	**3.965 ± 0.524**	**0.913 ± 0.131**	**0.957 ± 0.046**
Ideal value	**0**	**0**	**1**	**1**

in these datasets having four instead of eight bands, we only need to adjust the number of input channels for the first convolutional layer and the number of output channels for the last convolutional layer. After that, the adjusted network is trained on the GF-2 and QB training sets. Figs. 12 and 13 show the visual results of some representative and recent CNN-based methods. It is clear that the CMLNet has fewer residuals. As shown in Table V, the CMLNet gets the best quantitative results.

H. Ablation Studies

Our CMLNet consists of two key components: the multiplication network and the CML-resblock. The experimental results indicate that the combination of these two key components, i.e., the proposed CMLNet, gets superior performance. To separately show the superiority of these two key components, the results from different combinations of network

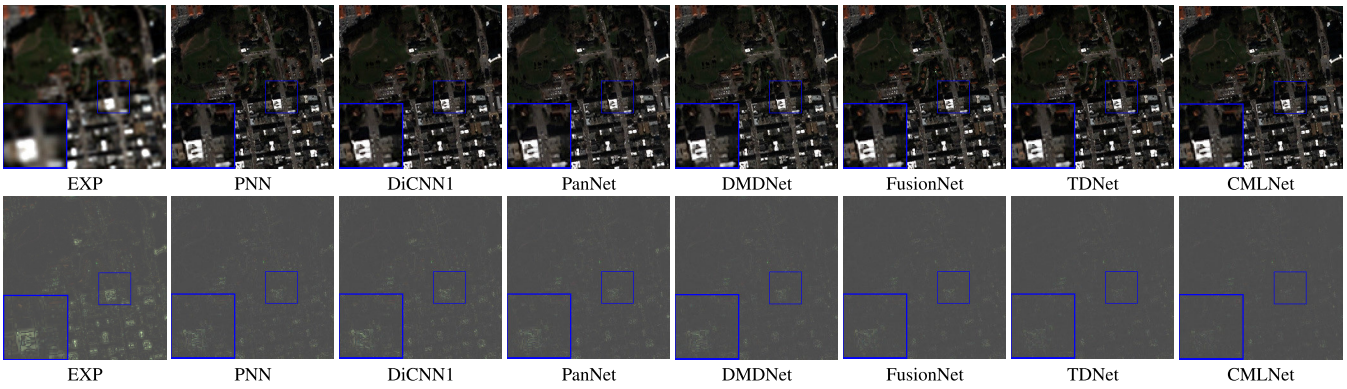


Fig. 12. Visual comparison among some representative and state-of-the-art CNN-based approaches on the Indianapolis dataset (sensor: QB). The corresponding AEMs using the GT image are depicted in the second row. For better visualization, we doubled the intensities of the AEMs and added 0.3.

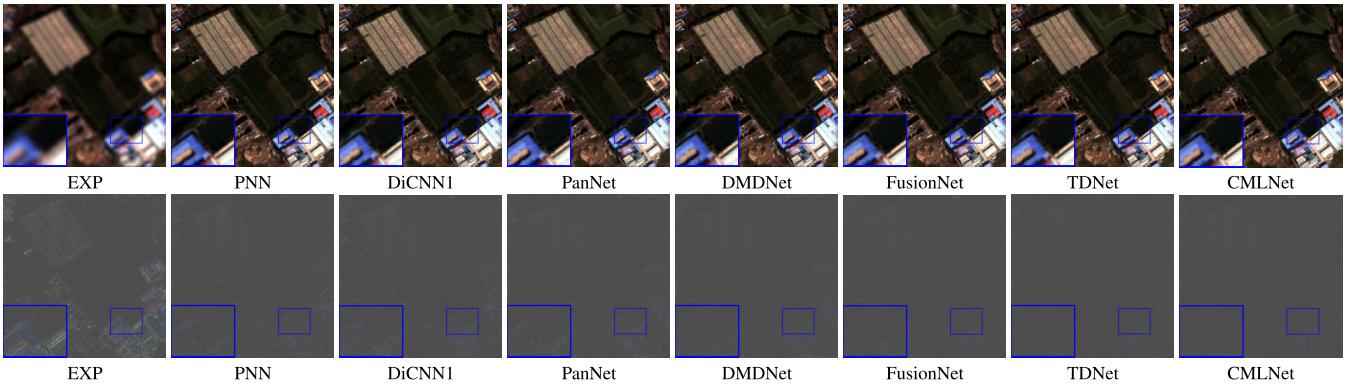


Fig. 13. Visual comparison among some representative and state-of-the-art CNN-based approaches on the Guangzhou dataset (sensor: GF-2). The corresponding AEMs using the GT image are depicted in the second row. For better visualization, we doubled the intensities of the AEMs and added 0.3.

TABLE VI

TUNING PARAMETERS OF THE COMPARED MODULES INCLUDED INTO THE NETWORK BACKBONE. NOTATION: **FILT. #** (NUMBER OF FILTERS FOR EACH LAYER), **N** (NUMBER OF BLOCKS), **LY. #** (NUMBER OF CONVOLUTION LAYERS IN THE BACKBONE), C (NUMBER OF FEATURE CHANNELS IN EACH GROUP), G (NUMBER OF CONVOLUTION KERNEL GROUPS), W (NUMBER OF FEATURE CHANNELS IN EACH SUBSET), S (NUMBER OF SUBSETS)

Module	Resblock	Bottleneck	Inception V3	Standard	Res2Net	CML-resblock
Filt. #	32c	72c	64c	4c×18g	18w×4s	4c×18g
N	4	4	4	4	4	4
Ly. #	8	12	12	20	20	20

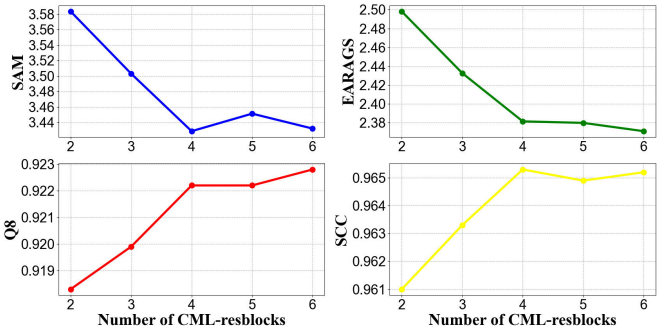


Fig. 14. Number of CML-resblocks stacked in the backbone of the proposed CMLNet against the average quality metrics on 1258 reduced-resolution samples for WV-3.

architectures and backbone modules are discussed. These experiments are obtained considering the WV-3 dataset.

We chose PanNet [1], FusionNet [2], and the multiplication network as network structures for comparison. Instead, as backbone modules, the resblock, the bottleneck block [36], the Inception-A block [45], the standard convolution, the Res2Net module, and the CML-resblock have been selected. The tuning parameters of the different modules are shown in Table VI.

1) Comparison Among Structures: To compare the performance of the different structures, we fix the backbone module. The results in Table VII show that the multiplication network overcomes PanNet and FusionNet in all the cases except for the resblock, thus demonstrating the effectiveness of the proposed multiplicative scheme.

2) Comparison Among Modules: From Table VII, it is clear to note that the standard convolution, the Res2Net, and the CML-resblock have a better performance compared with the resblock and the bottleneck block when embedded into different network structures. We can attribute this to an increase in the number of convolutional layers as shown in Table VI. Hence, the related networks can capture richer and more complex features.

Besides, despite Res2Net extracts multiscale information, its performance is similar to the one of the standard convolution when embedded into the PanNet and into the multiplication network. On the other hand, the CML-resblock gets the highest

TABLE VII

QUALITY METRICS FOR ALL THE COMPARED COMBINATIONS ON THE REDUCED-RESOLUTION DATASETS. WE COMPARE THEM WITHIN GROUPS USING THE SAME MODULE. (BOLD: BEST)

Module	Structure	*(a) Rio dataset*				*(b) Tripoli dataset*				*(c) 1258 samples*			
		SAM↓	*ERGAS*↓	*Q8*↑	*SCC*↑	*SAM*↓	*ERGAS*↓	*Q8*↑	*SCC*↑	*SAM*↓	*ERGAS*↓	*Q8*↑	*SCC*↑
Resblock	**PanNet**	4.8500	3.1744	0.9190	0.9642	4.6079	3.4227	0.9395	0.9516	4.0921	2.9524	0.8941	0.9494
	FusionNet	**4.3850**	**2.8533**	0.9422	**0.9718**	**4.2764**	**3.0568**	**0.9522**	**0.9646**	**3.7435**	**2.5679**	**0.9135**	**0.9580**
	Multiplication	4.5300	2.9266	**0.9429**	0.9696	4.3103	3.1268	0.9509	0.9620	3.8496	2.6760	0.9112	0.9549
Bottleneck	**PanNet**	4.9182	3.2812	0.9211	0.9599	4.9204	3.6873	0.9285	0.9387	4.2151	3.0597	0.8866	0.9466
	FusionNet	4.3943	2.8750	0.9450	0.9717	4.2420	3.0980	0.9522	0.9633	3.8369	2.6836	0.9100	0.9552
	Multiplication	**4.1830**	**2.7599**	**0.9480**	**0.9751**	**4.0307**	**2.9332**	**0.9563**	**0.9687**	**3.6531**	**2.5482**	**0.9131**	**0.9598**
Inception V3	**PanNet**	4.4187	2.9154	0.9298	0.9719	4.3875	3.2195	0.9455	0.9597	3.9930	2.9068	0.8976	0.9520
	FusionNet	4.1851	2.7315	0.9498	0.9753	4.0307	2.9351	0.9568	0.9681	3.6588	2.5517	0.9142	0.9596
	Multiplication	**4.0995**	**2.6837**	**0.9490**	**0.9769**	**3.9010**	**2.8247**	**0.9595**	**0.9719**	**3.5816**	**2.4814**	**0.9184**	**0.9616**
Standard	**PanNet**	4.4011	2.8857	0.9304	0.9731	4.2485	3.0891	0.9516	0.9645	3.9889	4.2109	0.8953	0.9424
	FusionNet	4.2643	2.8112	0.9464	0.9730	4.1003	3.0068	0.9522	0.9665	3.7224	2.6530	0.9087	0.9583
	Multiplication	**4.1211**	**2.7097**	**0.9505**	**0.9757**	**3.9497**	**2.8527**	**0.9584**	**0.9704**	**3.4959**	**2.4277**	**0.9201**	**0.9637**
Res2Net	**PanNet**	4.4073	2.8991	0.9284	0.9723	4.2834	3.0929	0.9479	0.9644	4.0097	2.9174	0.8934	0.9524
	FusionNet	4.1952	2.7628	0.9478	0.9748	4.0252	2.9538	0.9563	0.9682	3.6613	2.5509	0.9144	0.9596
	Multiplication	**4.1189**	**2.6931**	**0.9513**	**0.9767**	**3.9207**	**2.8133**	**0.9593**	**0.9722**	**3.4872**	**2.4122**	**0.9215**	**0.9638**
CML-resblock	**PanNet**	4.2682	2.8446	0.9309	0.9748	4.1223	2.9986	0.9540	0.9683	3.8976	2.9318	0.8991	0.9518
	FusionNet	4.1546	2.8083	0.9468	0.9744	4.0613	3.0047	0.9541	0.9679	3.7169	2.6748	0.9110	0.9595
	Multiplication	**4.0025**	**2.6217**	**0.9532**	**0.9783**	**3.7900**	**2.7091**	**0.9625**	**0.9744**	**3.4285**	**2.3815**	**0.9223**	**0.9653**
Ideal value		**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**

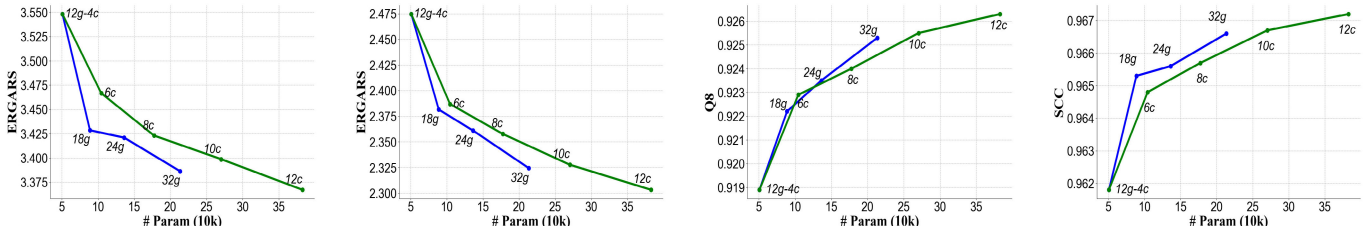


Fig. 15. Size of convolution kernels by changing the number of groups (g; blue line) or channel (c; green line) within each group against the average quality metrics on 1258 reduced-resolution samples for WV-3.

performance, showing that the multiscale extraction of the Res2Net is insufficient and unbalanced with respect to the proposed one.

V. DISCUSSION

In this section, we analyze the computational complexity and the performance of the proposed CMLNet by changing the number of CML-resblocks and the size of convolution kernels in the CML-resblock. Finally, we discuss the strengths and weaknesses which could be focused on for further research.

A. Number of CML-Resblocks

According to the results in Fig. 14, we observe that the performance of the network tends to improve with the increase in the number of the CML-resblocks. This can be attributed to the increase in the NoPs, which improves the nonlinear ability of the network. It is worth noting that the results obtained by a network with four stacked convolutional layers represent a good tradeoff between performance and computational burden.

B. Size of Convolution Kernels

We divide the kernels of a convolutional layer into 18 groups with four channels in each group. In this section, we test the impact of using different sizes for the convolution kernels changing the number of groups (g) and channels (c) in each group when setting the number of CML-resblocks to four.

When the number of convolution kernels grows, the features extracted become more and more diverse although there can be some similar features. The experimental results in Fig. 15 show that the number of groups and channels in each group can improve the network performance. Moreover, considering the NoPs, the increase in the number of groups can lead to more gains, thus demonstrating the need to use group convolutions in our CMLNet.

C. Future Perspectives

Based on the previously shown results, CMLNet has proved its advantages in coping with the pansharpening problem. Furthermore, there are also some issues that could be developed. On one hand, CML-resblock provides pixel-level multireceptive learning ability through a cascadic connection strategy, but it needs more parameters and execution time than other methods with the same number of middle-layer channels. Besides, we did not demonstrate the effectiveness of CML-resblock in other tasks. On the other hand, although the proposed network structure is based on the HPM (or multiplicative)

injection scheme which generally performs better than the additive injection model, we only simply stack some modules in the backbone to learn the coefficients to prove its simplicity and effectiveness. Thus, it could be regarded as a preliminary examination, and we can design specific backbone modules considering more properties of the PAN and LRMS images.

VI. Conclusion

In this article, we introduced a new architecture to solve the pansharpening problem, called multiplication network. Unlike other methods that learn spatial details based on the additive injection model, our multiplication network learns coefficients for a restoration mapping function that is used to multiply the upsampled low spatial resolution multispectral image to generate the target image. To further enhance the feature extraction ability of our method, we designed the CML-resblock module based on ResNet and Res2Net, which can effectively extract multiscale information. Afterward, we incorporated the CML-resblock into the backbone of our multiplication network to design the overall network, i.e., the CMLNet. The experiments on both the reduced- and full-resolution datasets demonstrate the effectiveness of the proposed CMLNet. Furthermore, ablation studies confirm the superiority of both our multiplication network architecture and the CML-resblock. Finally, the impacts of the number of CML-resblocks stacked into the network backbone and the convolution kernel size have been discussed.

References

[1] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "PanNet: A deep network architecture for pan-sharpening," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1753–1761.

[2] L.-J. Deng, G. Vivone, C. Jin, and J. Chanussot, "Detail injection-based deep convolutional neural networks for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6995–7010, Aug. 2021.

[3] G. Vivone et al., "A new benchmark based on recent advances in multispectral pansharpening: Revisiting pansharpening with classical and emerging pansharpening methods," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 53–81, Mar. 2021.

[4] G. Vivone, P. Addesso, R. Restaino, M. D. Mura, and J. Chanussot, "Pansharpening based on deconvolution for multiband filter estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 540–553, Jan. 2019.

[5] G. Vivone, R. Restaino, G. Licciardi, M. D. Mura, and J. Chanussot, "Multiresolution analysis and component substitution techniques for hyperspectral pansharpening," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 2649–2652.

[6] L. Loncan et al., "Hyperspectral pansharpening: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 27–46, Sep. 2015.

[7] G. Vivone, "Multispectral and hyperspectral image fusion in remote sensing: A survey," *Inf. Fusion*, vol. 89, pp. 405–417, Jan. 2023.

[8] G. Vivone et al., "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2565–2586, May 2015.

[9] X. Meng, H. Shen, H. Li, L. Zhang, and R. Fu, "Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges," *Inf. Fusion*, vol. 46, pp. 102–113, Mar. 2019.

[10] B. Aiazzi, L. Alparone, A. Arienzo, A. Garzelli, and S. Lolli, "Fast multispectral pansharpening based on a hyper-ellipsoidal color space," *Proc. SPIE*, vol. 11155, Oct. 2019, Art. no. 1115507.

[11] Q. Xu, Y. Zhang, B. Li, and L. Ding, "Pansharpening using regression of classified MS and pan images to reduce color distortion," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 28–32, Jan. 2015.

[12] A. Garzelli, F. Nencini, and L. Capobianco, "Optimal MMSE pan sharpening of very high resolution multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 1, pp. 228–236, Jan. 2008.

[13] G. Vivone, "Robust band-dependent spatial-detail approaches for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6421–6433, Sep. 2019.

[14] J. Choi, K. Yu, and Y. Kim, "A new adaptive component-substitution-based satellite image fusion by using partial replacement," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 295–309, Jan. 2011.

[15] C. A. Laben and B. V. Brower, "Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening," U.S. Patent 6 011 875, Jan. 4, 2000.

[16] S. Lolli, L. Alparone, A. Garzelli, and G. Vivone, "Haze correction for contrast-based multispectral pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2255–2259, Dec. 2017.

[17] J. G. Liu, "Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details," *Int. J. Remote Sens.*, vol. 21, no. 18, pp. 3461–3472, Jan. 2000.

[18] X. Otazu, M. Gonzalez-Audicana, O. Fors, and J. Nunez, "Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 10, pp. 2376–2385, Oct. 2005.

[19] B. Aiazzi, L. Alparone, S. Baronti, and A. Garzelli, "Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2300–2312, Jan. 2002.

[20] G. Vivone, R. Restaino, M. D. Mura, G. Licciardi, and J. Chanussot, "Contrast and error-based fusion schemes for multispectral image pansharpening," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 5, pp. 930–934, May 2014.

[21] G. Vivone, R. Restaino, and J. Chanussot, "Full scale regression-based injection coefficients for panchromatic sharpening," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3418–3431, Jul. 2018.

[22] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé, "A variational model for P+XS image fusion," *Int. J. Comput. Vis.*, vol. 69, no. 1, pp. 43–58, Aug. 2006.

[23] P. Guo, P. Zhuang, and Y. Guo, "Bayesian pan-sharpening with multiorder gradient-based deep network constraints," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 950–962, 2020.

[24] Y. Yang, L. Wu, S. Huang, W. Wan, W. Tu, and H. Lu, "Multiband remote sensing image pansharpening based on dual-injection model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1888–1904, 2020.

[25] M. R. Vicinanza, R. Restaino, G. Vivone, M. D. Mura, and J. Chanussot, "A pansharpening method based on the sparse representation of injected details," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 180–184, Jan. 2015.

[26] Z. C. Wu, T. Z. Huang, L. J. Deng, and G. Vivone, "A framelet sparse reconstruction method for pansharpening with guaranteed convergence," *Inverse Problems Imag.*, vol. 17, no. 6, pp. 1277–1300, 2023.

[27] L.-J. Deng, G. Vivone, W. Guo, M. D. Mura, and J. Chanussot, "A variational pansharpening approach based on reproducible kernel Hilbert space and heaviside function," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4330–4344, Sep. 2018.

[28] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, J. Huang, J. Chanussot, and G. Vivone, "LRTCFPan: Low-rank tensor completion based framework for pansharpening," *IEEE Trans. Image Process.*, vol. 32, pp. 1640–1655, 2023.

[29] X. Cao, X. Fu, D. Hong, Z. Xu, and D. Meng, "PanCSC-Net: A model-driven deep unfolding method for pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5404713.

[30] L.-J. Deng et al., "Machine learning in pansharpening: A benchmark, from shallow to deep networks," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 3, pp. 279–315, Sep. 2022.

[31] S.-Q. Deng, L.-J. Deng, X. Wu, R. Ran, D. Hong, and G. Vivone, "PSRT: Pyramid shuffle-and-reshuffle transformer for multispectral and hyperspectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5503715.

[32] Y.-W. Zhuo, T.-J. Zhang, J.-F. Hu, H.-X. Dou, T.-Z. Huang, and L.-J. Deng, "A deep-shallow fusion network with multidetail extractor and spectral attention for hyperspectral pansharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 7539–7555, 2022.

[33] R. Wen, L.-J. Deng, Z.-C. Wu, X. Wu, and G. Vivone, "A novel spatial fidelity with learnable nonlinear mapping for panchromatic sharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5401915.

[34] R. Ran, L.-J. Deng, T.-X. Jiang, J.-F. Hu, J. Chanussot, and G. Vivone, "GuidedNet: A general CNN fusion framework via high-resolution guidance for hyperspectral image super-resolution," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4148–4161, Jul. 2023.

[35] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sens.*, vol. 8, no. 7, p. 594, Jul. 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] Y. Zhang, W. Li, M. Zhang, Y. Qu, R. Tao, and H. Qi, "Topological structure and semantic information transfer network for cross-scene hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 2817–2830, Jun. 2023.

[38] M. Zhang, X. Zhao, W. Li, Y. Zhang, R. Tao, and Q. Du, "Cross-scene joint classification of multisource data with multilevel domain adaption network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 6, 2023, doi: 10.1109/TNNLS.2023.3262599.

[39] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion," *Inf. Fusion*, vol. 62, pp. 110–120, Oct. 2020.

[40] S. Luo, S. Zhou, Y. Feng, and J. Xie, "Pansharpening via unsupervised convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4295–4310, 2020.

[41] Z. Xiong, N. Liu, N. Wang, Z. Sun, and W. Li, "Unsupervised pansharpening method using residual network with spatial texture attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023.

[42] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 4905–4913.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[44] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[47] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[48] L. Alparone, A. Garzelli, and G. Vivone, "Intersensor statistical matching for pansharpening: Theoretical issues and practical solutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4682–4695, Aug. 2017.

[49] G. Vivone, R. Restaino, and J. Chanussot, "A regression-based high-pass modulation pansharpening approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 984–996, Feb. 2018.

[50] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, vol. 63, no. 6, pp. 691–699, 1997.

[51] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "MTF-tailored multiscale fusion of high-resolution MS and pan imagery," *Photogram. Eng. Remote Sens.*, vol. 72, no. 5, pp. 591–596, May 2006.

[52] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. M. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data-fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3012–3021, Oct. 2007.

[53] L. He et al., "Pansharpening via detail injection based convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1188–1204, Apr. 2019.

[54] X. Fu, W. Wang, Y. Huang, X. Ding, and J. Paisley, "Deep multiscale detail networks for multiband spectral image sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2090–2104, May 2021.

[55] T.-J. Zhang, L.-J. Deng, T.-Z. Huang, J. Chanussot, and G. Vivone, "A triple-double convolutional neural network for panchromatic sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9088–9101, Nov. 2023.

[56] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. JPL Airborne Geosci. Workshop, AVIRIS Workshop*, Pasadena, CA, USA, 1992, pp. 147–149.

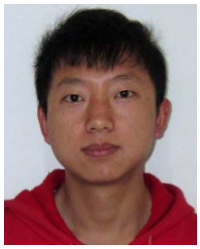
[57] L. Wald, *Data Fusion: Definitions and Architectures: Fusion of Images of Different Spatial Resolutions*. Paris, France: Presses des MINES, 2002.

[58] J. Zhou, D. L. Civco, and J. A. Silander, "A wavelet transform method to merge Landsat TM and SPOT panchromatic data," *Int. J. Remote Sens.*, vol. 19, no. 4, pp. 743–757, Jan. 1998.

[59] A. Garzelli and F. Nencini, "Hypercomplex quality assessment of multi/hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 662–665, Oct. 2009.

[60] A. Arienzo, G. Vivone, A. Garzelli, L. Alparone, and J. Chanussot, "Full-resolution quality assessment of pansharpening: Theoretical and hands-on approaches," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 3, pp. 168–201, Sep. 2022.

Jun-Da Wang is currently a Senior Undergraduate of Mathematics and Physics Fundamental Science with the Yingcai Honors College, University of Electronic Science and Technology of China (UESTC), Chengdu, China, supervised by Prof. Liang-Jian Deng. His research interests include machine learning and deep learning for image processing and image fusion.



Liang-Jian Deng (Senior Member, IEEE) received the B.S. and Ph.D. degrees in applied mathematics from the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2016, respectively.

He is currently a Professor with the School of Mathematical Sciences, UESTC. From 2013 to 2014, he was a joint-training Ph.D. Student with the Case Western Reserve University, Cleveland, OH, USA. In 2017, he was a Post-Doctoral Researcher with Hong Kong Baptist University (HKBU), Hong Kong. In addition, he also stayed at the Isaac Newton Institute for Mathematical Sciences, Cambridge University, Cambridge, U.K., and HKBU, for short visits. His research interests include the use of partial differential equations (PDEs), optimization modeling, and deep learning to address several tasks in image processing and computer vision, e.g., resolution enhancement and restoration.

Chen-Yu Zhao received the B.S. degree in communication engineering from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2023. She is currently pursuing the M.Phil. degree with the School of Computer Science Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong.

Her research interests include machine learning, deep learning in image processing, and medical image analysis.

Xiao Wu received the B.S. degree in intelligent science and technology from the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2019, and the M.S. degree from the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2023, where he is currently pursuing the Ph.D. degree under Prof. Ting-Zhu Huang.

His research interests include machine learning and deep learning for computer vision, image processing, and image fusion.

Hong-Ming Chen received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from National Tsing Hua University, Hsinchu, China, in 1997 and 2002, respectively, and the Ph.D. degree in microelectronics engineering from Peking University, Beijing, China, in 2015.

He has worked for Shanghai Taolink Technologies Corporation, Shanghai, China; SiFive, San Mateo, CA, USA; Global Unichip Corporation, Hsinchu, Taiwan; and Faraday Tech., Hsinchu, in the area of Internet of Things (IoT), RISC-V processor, high-speed, and low-power System On Chip (SoC) design. He is currently a Professor with the School of Information Engineering, Zhejiang Ocean University. He is also currently a Long Term Part-Time Professor with the School of Electronic Information, Wuhan University, and the School of Electronics Information and Optical Engineering, Nankai University. He has more than 20 years of experience in semiconductor, data communication, and networking infrastructure industries. His research interests include ocean exploration, carbon neutralization, IoT, RISC-V processor, Artificial intelligence (AI) accelerator, high-performance computing, and high-speed networking.

Gemine Vivone (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees (summa cum laude), and the Ph.D. degree in information engineering from the University of Salerno, Fisciano, Italy, in 2008, 2011, and 2014, respectively.

He is currently a Researcher with the National Research Council—Institute of Methodologies for Environmental Analysis (CNR-IMAA), Tito Scalo, Italy, and National Biodiversity Future Center (NBFC), Palermo, Italy. His main research interests include statistical signal processing, detection of remotely sensed images, data fusion, and tracking algorithms.