PSRT: Pyramid Shuffle-and-Reshuffle Transformer for Multispectral and Hyperspectral Image Fusion

Shang-Qi Deng, Liang-Jian Deng, *Member, IEEE*, Xiao Wu, Ran Ran, Danfeng Hong, *Senior Member, IEEE*, and Gemine Vivone, *Senior Member, IEEE*

***Abstract*—Transformer has received a lot of attention in computer vision. Because of global self-attention, the computational complexity of Transformer is quadratic with the number of tokens, leading to limitations for practical applications. Hence, the computational complexity issue can be efficiently resolved by computing the self-attention in groups of smaller fixed-size windows. In this paper, we propose a novel Pyramid Shuffle-and-Reshuffle Transformer (PSRT) for the task of Multispectral and Hyperspectral Image Fusion (MHIF). Considering the strong correlation among different patches in remote sensing images and complementary information among patches with high similarity, we design Shuffle-and-Reshuffle (SaR) modules to consider the information interaction among global patches in an efficient manner. Besides, using pyramid structures based on window self-attention, the detail extraction is supported. Extensive experiments on four widely-used benchmark datasets demonstrate the superiority of the proposed PSRT with a few parameters compared with several state-of-the-art approaches. The related code will be available soon.**

***Index Terms*—Multispectral and Hyperspectral Image Fusion, Shuffle-and-Reshuffle Transformer, Pyramid Structure, Image Enhancement, Image Fusion, Remote Sensing.**

I. Introduction

Multispectral and Hyperspectral Image Fusion (MHIF) [2]–[13], [13]–[19] is a classical task in computer vision involving high spectral resolution hyperspectral data, which have a limited spatial resolution because of some physical constraints. MHIF aims to generate a High-Resolution Hyperspectral Image (HR-HSI) by fusing a High-Resolution Multispectral Image (HR-MSI) and a Low-Resolution HSI (LR-HSI). These outcomes can be used for object recognition [20], classification [21]–[25], and segmentation [26], [27]. Despite many efforts that have recently been made [2], [14], [28]–[37], the design of a high-efficiency technology for the problem at hand is still a challenging task.

Convolutional Neural Networks (CNNs) shine in the field of computer vision thanks to their high accuracy. While convolution operations have been extensively analyzed and exploited, well-known drawbacks include redundancy and spatial-agnostic [38]–[42] in convolutional kernels [43], [44]. In particular, convolutional kernels are shared in different positions of the image. Hence, the results are region-independent, leading to the difficulty of capturing long-distance dependency in the feature map. Recently, Transformers [45] have performed well in many computer vision fields, mainly due to their powerful ability to characterize long-distance relationships that are of crucial importance for MHIF.

Transformer has been extensively used with the aim of modeling non-local relationships in images. In prior works, Vision Transformer (ViT) [46] considered the use of a Transformer in Natural Language Processing (NLP) for image classification. It divides the image into fixed-size patches, then embeds them into a linear layer. Similarly to the processing of Transformers in NLP, patches are regarded as tokens, and then the self-attention mechanism [45] is performed on patches. The hier-

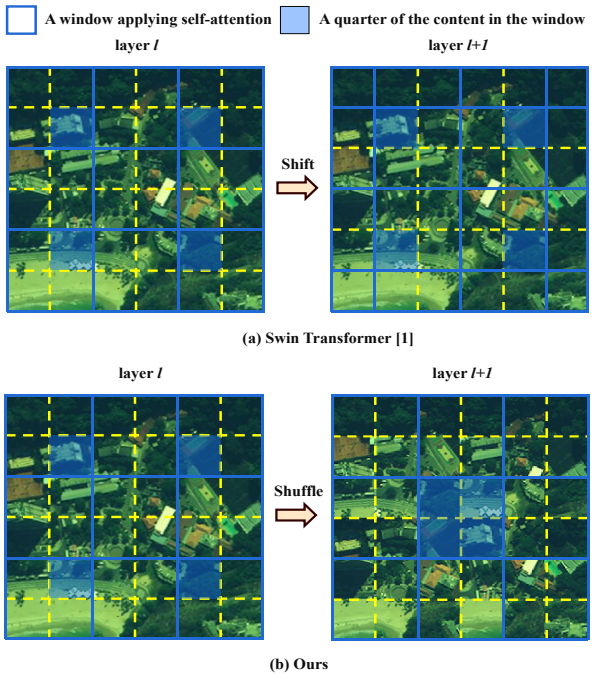


Fig. 1. The solid lines divide the plane into several fixed-size windows and the dotted grids are used to divide the original window into four equal parts. The blue block represents a quarter of the content in a window which has, for instance, 2×2 tokens. (a) The *Shifted Window* approach applied to the Swin Transformer [1] moving the blue blocks to the adjacent windows. (b) The *Shuffle* operation of our PSRT block for similarity calculation, which gathers the blue blocks of different windows into one window, realizing the propagation of local information to global regions by window self-attention.

This research is supported by NSFC (12271083, 42271350, 62203089), Natural Science Foundation of Sichuan Province (2022NSFSC0501, 2022NSFSC0507), and National Key Research and Development Program of China (Grant No. 2020YFA0714001). Corresponding Author: Liang-Jian Deng.

S.-Q. Deng, L.-J. Deng, X. Wu, and R. Ran are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China (e-mails: Shangqi_Deng@std.uestc.edu.cn; liangjian.deng@uestc.edu.cn; wxwsx1997@gmail.com).

D. Hong is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China. (hongdf@aircas.ac.cn).

G. Vivone is with the National Research Council - Institute of Methodologies for Environmental Analysis, CNR-IMAA, I-85050 Tito, Italy (e-mail: gemine.vivone@imaa.cnr.it).

archical framework of ViT inspired the use of a Transformer in the field of computer vision. Compared with ViT, Pyramid Vision Transformer (PVT) [47] introduced a pyramid structure exploited for segmentation and object detection. In this way, PVT can be used as a backbone like CNN in dense prediction tasks. Recently, Swin Transformer [1] demonstrated great potential and high efficiency. Like PVT, the Swin Transformer also adopted the hierarchical design, including four stages. Similar to [20], for expanding the receptive field in CNN, each stage reduces the resolution of the feature map. Window self-attention in the Swin Transformer empowers the model to handle large-sized images. The shifted window operation creates a long-range dependency between each independent window, and the masking realizes the calculation of attention for irregular windows without changing the original method. The development of high-level computer vision tasks has also led to the innovation of low-level tasks. Indeed, SwinIR [48] borrowed the structure of the Swin Transformer, achieving competitive performance in low-level tasks, such as image super-resolution [49], image de-raining [50], [51], image denoising [52], and JPEG compression [53]. SwinIR fully exploits the advantages of Swin Transformer and designs cascade blocks for deep feature extraction.

In this paper, we introduce the Pyramid Shuffle-and-Reshuffle Transformer (PSRT), a new stage-to-stage hierarchical framework for MHIF designing a novel way for information interaction. In summary, the contributions of this paper are as follows:

- We propose the so-called PSRT for the task of MHIF, combining the Shuffle-and-Reshuffle (SaR) strategy and the multi-scale feature extraction to learn both local and more distant representations, and reducing the amount of computation compared with the ViT.
- The customized SaR strategy can propagate the information among different windows and enhance the efficiency of modeling long-distance dependence. Besides, the design of the window pyramid structure can capture features at different granularities (resolutions), recovering, in a suitable way, detailed information for MHIF.
- We demonstrate the performance of the proposed approach on four commonly used datasets, i.e., Chikusei, CAVE, Harvard, and Pavia. Results show that the proposed approach can achieve state-of-the-art performance with fewer parameters.

The rest of the paper is organized as follows. In Sect. II, we introduce the related works about the MHIF problem. Sect. III presents the proposed SaR strategy as well as the network architecture. In Sect. IV, extensive experiments are conducted to assess the effectiveness of the proposed SaR strategy and PSRT block. In addition, experiments on the performance of the SaR strategy considering image boundaries and a comparison with the shifted window approach in [1] are also provided to the readers.

II. Related Works

A. CNNs and MHIF

Very recent attempts to solve the MHIF task are often based on the use of CNNs. SSRNet [54] proposed a physical straightforward CNN model designing two loss functions for spatial and spectral reconstructions, respectively. The residual mechanism [55] is widely used to optimize the structure of networks, and approaches like ResTFNet [56] exploited it to avoid model degradation caused by the deep network. MHFNet [57] employed the convolutional expansion optimization algorithm to obtain a new network with the aim of improving interpretability. MoG-DCN [58] adopted a U-Net [59] deep convolutional network (DCN), which can exploit the multi-scale dependence of HSIs. After model-guided unrolling, the entire network is trained end-to-end using a DCN-based denoiser. HSRNet [30] exploited channel attention [60] and spatial attention modules to extract information from different dimensions. Although CNN is a powerful structure, the use of Transformers demonstrated to have a great potential for computer vision.

B. Self-Attention in MHIF

The core module for the Transformer is the self-attention mechanism. Unlike convolutional operations in CNNs, the self-attention mechanism can theoretically expand the receptive field infinitely, thereby correlating different patches with each other. However, applying self-attention directly to the feature map in a pixel-to-pixel fashion leads to a dramatic increase in the computation burden. ViT subtly splits the feature maps into fixed-size patches, linearly embedding each of them and feeding the sequence of the resulting vectors to a standard Transformer encoder. PVT inherits the advantages of both CNN and Transformer, making a unified backbone for various vision tasks and directly replacing the CNN backbone. PVT differs from ViT because it can be trained on dense partitions of images to achieve high-resolution outputs, and it leverages on a stage-to-stage structure to reduce the computation. In T2T-ViT [61], novel tokenization for ViT has been developed, in which the adjacent tokens are further processed by splicing to achieve the purpose of aggregating information and reducing parameters. In addition, researchers also attempted to apply Transformer to low-level tasks. For instance, the SwinIR [48] approach relies upon a robust model for image restoration where the structure of Swin Transformer has been directly used to build a Transformer block for deep feature extraction.

C. Shuffle-wise Operation

In recent years, works as in [62], [63] addressed the problem of creating a cross-window connection through shuffle operation. For easier understanding, it is assumed that the input is a 1D sequence to preserve generality, and a Window-based Self-Attention with a window size of M and an input of N tokens is considered. Shuffle Transformer [62] uses Self-Attention first, reshapes the spatial input into $(M, \frac{N}{M})$, transposes, and then flats it to serve it as input for the Window-based Self-Attention. Long-range cross-window connections are made

possible by this type of operation, which groups tokens from several windows. Ocnet [63] is based on two stages, i.e., local and global, to implement the cross-window connection. The first stage uses window-based Self-Attention to deal with local information, and the other stage reshapes the input spatial into ($\frac{N}{M}$, M), transposes, and then flats it to serve it as input for the Window-based Self-Attention. This latter approach shows a crucial flaw, that is, the number of windows $\frac{N}{M}$ is quite high if the window size, M, in the first layer is small.

D. Motivation

There are many similar patches inside a hyperspectral image, all closely related. Through the ViT model, patches in different positions can be associated with each other to achieve the purpose of restoring image details, but the computational cost of this method is quadratic with inputs. The method of window self-attention solves the problem of high computation but limits the ability to model long-distance dependence. Swin Transformer [1] solves these problems through the following steps: 1) the original windows are shifted a half window size to get new windows; moreover, some newly generated windows are made up with original quarter and half windows at the image boundary; 2) a mask operation is exploited to independently compute self-attention for the original quarter and half windows; 3) the computed windows are re-shifted. In Fig. 1-(a), we can observe that the size of the windows close to the image boundaries becomes smaller after the shifting and masking operations. Thus, the mask forces a part of the attention matrix to negative numbers, leading to an insufficient self-attention calculation for the windows located at image boundaries. This issue weakens the information interaction at image boundaries, thus motivating us to improve Swin Transformer through the proposed SaR strategy (which does not use any mask to realize global attention). Fig. 1-(b) shows the state of the windows after using the proposed SaR strategy. By shuffling the plane according to the given rules and then implementing window self-attention, a long-distance connection can be achieved to obtain a global correlation in a quick way. The multi-scale design is of crucial importance for MHIF, and the window attention is the greatest advantage of the Swin Transformer. Thus, we developed a multi-scale window attention (i.e., the given PSRT block) inspired by the classical pyramid structure to enrich features and yield better information interaction. Sects.IV-H and IV-J assess the effectiveness of the proposed approach.

III. Methodology

This section is devoted to the introduction of the main blocks and mechanisms of the proposed approach together with the adopted loss function.

A. Overview

As illustrated in Fig. 5, PSRT follows the commonly used hierarchical architecture, which concatenates HR-MSI and the upsampled version of the LR-HSI, and the whole architecture learns the residual between the upsampled version of the LR-HSI and the ground-truth (GT). The first convolution layer extracts the shallow features and lifts the number of channels. Then, we use PSRT blocks to extract deep information, in which each PSRT block has a decreasing size of the window for self-attention in order to extract information at different scales. Finally, a convolutional layer is used as decoder to merge the information. The outcome is obtained by adding the upsampled version of the LR-HSI to the output.

B. Window Self-Attention

Transformer is characterized by the establishment of long-distance dependence among tokens, which can effectively describe the global correlation. For an image, we usually take a 4×4 patch as a token and then inputting it into the attention mechanism for calculation. Although this alleviates the computational problem to a certain extent, it does not solve the problem that this approach is expensive due to the use of the original size as input. To address this issue, we divide the input image into smaller non-overlapped windows in the spatial dimension (i.e., Swin Transformer [1]), then we implement self-attention to the tokens in each window for efficient calculation. The window self-attention can be expressed as follows:

$$\mathrm{P}_i = \sum_{i,j \in \Omega} \delta_{i \to j} \mathrm{V}_j, \tag{1}$$

where $\Omega = \{1, \cdots, N\}$ represents a collection of indexes within a separate window that contains a set of N tokens and P_i stands for the i-th token after the attention computation and matrix multiplication in Ω. We project the N tokens in Ω into three parts: query, key, and value through matrix multiplication. The projection can be expressed as follows:

$$\begin{aligned} \mathrm{Q} &= \mathrm{X}\mathrm{P}_{\mathrm{Q}}, \\ \mathrm{K} &= \mathrm{X}\mathrm{P}_{\mathrm{K}}, \\ \mathrm{V} &= \mathrm{X}\mathrm{P}_{\mathrm{V}}, \end{aligned} \tag{2}$$

where $\mathrm{P_Q}$, $\mathrm{P_K}$, and $\mathrm{P_V}$ represent the projected matrices that are shared across different windows.

The attention result of the query from the i-th key to the j-th key is marked as $\delta_{i \to j}$, V_j is the value of the j-th token and $\delta_{i \to j}$ is obtained by calculating the cosine value between Q_i and K_j via dot product, softmax function along the row, and a scaling factor. In this way, the complexity of self-attention decreases and it allows the model to gain the ability to handle large-sized feature maps and the original tokens can be expressed in a local manner.

C. Shuffle-and-Reshuffle (SaR) Strategy

*1) **Shuffle Operation**:* This section introduces the Shuffle operation, which consists of two parts, i.e., horizontal Shuffle and vertical Shuffle. The details of the two Shuffle operations are depicted in the flowchart in Fig. 2. More specifically, the horizontal Shuffle in the spatial domain can be described by the following equations:

$$\mathrm{G}_k = \left\{ \mathrm{V}_{ij} | k = \lfloor \frac{j}{d} \rfloor \right\}, \tag{3}$$

$$\mathrm{V} = \mathrm{Splice_h}[\mathrm{G}_1, \mathrm{G}_2, \cdots, \mathrm{G}_{\frac{W}{d}}], \tag{4}$$

$$\mathrm{V}_s = \mathrm{Splice_h}[\mathrm{G}_1, \mathrm{G}_3, \cdots, \mathrm{G}_{\frac{W}{d}-1}, \mathrm{G}_2, \mathrm{G}_4, \cdots, \mathrm{G}_{\frac{W}{d}}], \tag{5}$$

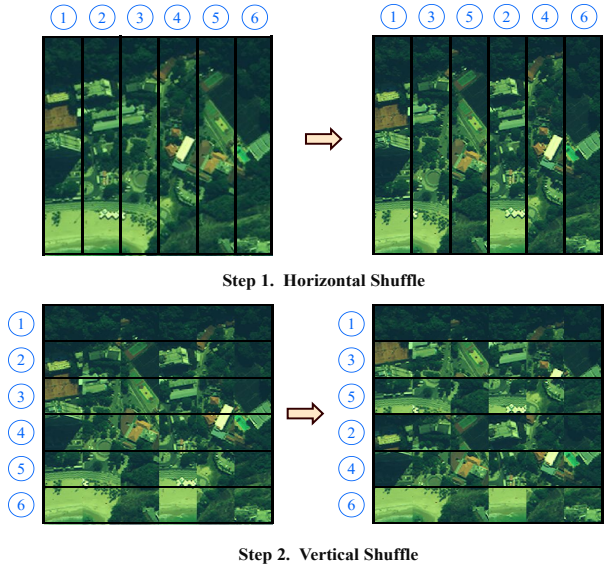


Fig. 2. The Shuffle consists of two parts, i.e., the Shuffle in the horizontal and vertical directions, respectively. Step 1 shows the Shuffle of the columns. Step 2 shows the Shuffle of the rows.

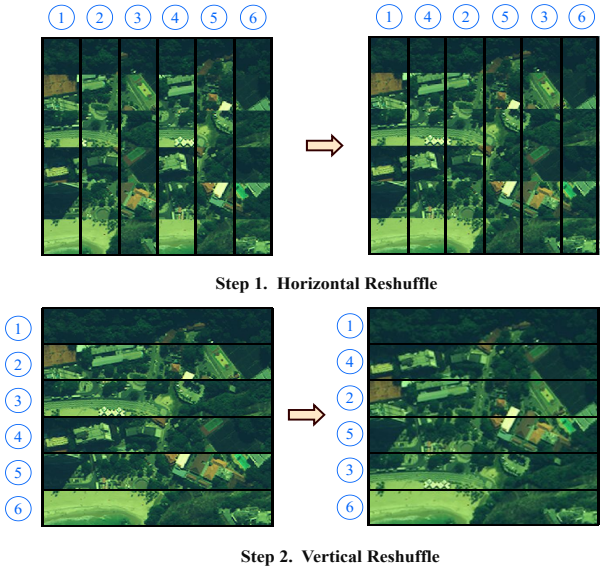


Fig. 3. The Reshuffle consists of two parts, i.e., the Reshuffle in the horizontal and vertical directions, respectively. Step 1 shows the Reshuffle of the columns. Step 2 shows the Reshuffle of the rows.

where $\lfloor \cdot \rfloor$ indicates the floor function.

For simplicity, we mark the plane as $V \in \mathbb{R}^{H \times W}$ and the tokens in the plane as $\mathrm{V}_{ij} \in \mathbb{R}$, $i \in \{1, 2, \cdots, H\}$, $j \in \{1, 2, \cdots, W\}$. In the above formula, d is the length of the Shuffle, whose value is $\frac{win\_size}{2}$, and, $\mathrm{Splice_h}$, in the above equations, is used to horizontally merge the groups G_k. Afterwards, we divide V into $\frac{W}{d}$ groups, denoted as G_k, $k \in \left\{1, 2, \cdots, \frac{W}{d}\right\}$. This operation is shown in the upper left corner of Fig. 2. Then, we perform the Shuffle operation on the plane V and the result of this operation is shown in the upper right corner of Fig. 2. After getting the horizontal Shuffle outcome, V_s, we exploit a similar strategy to V_s along the vertical direction to yield the final Shuffle result, see the second row of Fig. 2.

*2) **Reshuffle Operation**:* We execute the window self-attention to the shuffled tokens and reverse the obtained results by the Reshuffle operation, which still contains the horizontal and vertical directions. The specific operation can be found in Fig. 3 and the equations characterizing the operator along the horizontal direction are as follows:

$$\mathrm{G}_k = \left\{ \mathrm{V}_{ij} | k = \lfloor \frac{j}{d} \rfloor \right\}, \tag{6}$$

$$\mathrm{V}_s = \mathrm{Splice_h}[\mathrm{G}_1, \mathrm{G}_2, \cdots, \mathrm{G}_{\frac{W}{d}}], \tag{7}$$

$$\mathrm{V} = \mathrm{Splice_h}[\mathrm{G}_1, \mathrm{G}_{\frac{W}{2d}+1}, \mathrm{G}_2, \mathrm{G}_{\frac{W}{2d}+2} \cdots, \mathrm{G}_{\frac{W}{2d}}, \mathrm{G}_{\frac{W}{d}}], \tag{8}$$

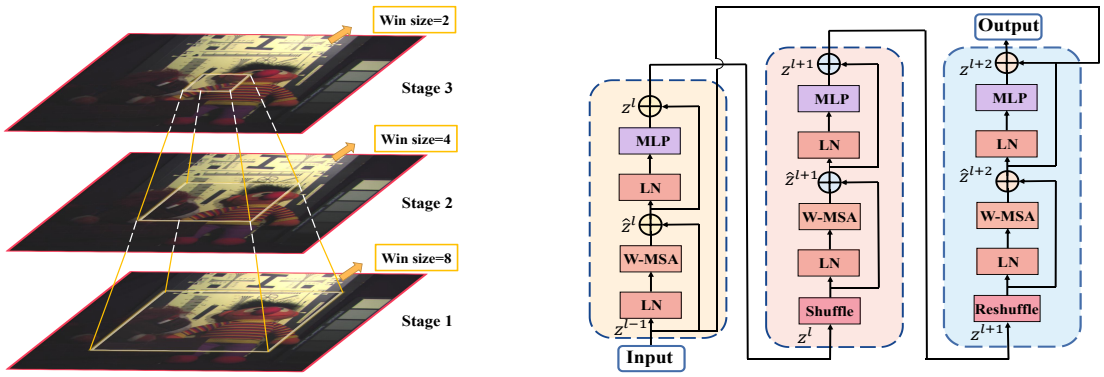
where V_s represents the plane that has been shuffled. According to Fig. 3, we first divide V_s into $\frac{W}{d}$ groups, denoted as G_k, $k \in \left\{1, 2, \cdots, \frac{W}{d}\right\}$, as in (7), and reshuffle the columns of V_s to get V by (8), then applying the Reshuffle operation in the vertical direction. The operation of Reshuffle is the inverse operation of Shuffle. By adding window self-attention between Shuffle and Reshuffle, the receptive field originally limited by the window can be expanded. The following section will detail the application of this module.

D. PSRT Block

In this section, we will detail the PSRT block that is added to the main network architecture as a block. The pyramid structure used in this work is not the same as the stage-to-stage structure like PVT [47] and Swin Transformer [1]. Our PSRT block has a window pyramid structure shown in Fig. 4-(a). It is not difficult to see that the size of the window decreases stage-to-stage to create multi-scale windows. In the PSRT block, the size of the window is fixed at each stage, and each window is independent of the others. The output of the previous stage is used as input for the next stage. In the meanwhile, the size of the window reduces to half of the previous stage. Specifically, a PSRT block consists of three stages, extracting different information through multi-scale windows in a hierarchical way, to recover local details. Especially, in the first stage, the size of input features is $H \times W \times C$ and the size of the window is $W_1 \times W_1$ (W_1 is equal to 8). When the features reach the last stage, the size of the window decays to $\frac{W_1}{4} \times \frac{W_1}{4}$, but the shape of the feature keeps unchanged. The multi-scale design is important for MHIF, and the window attention is the greatest advantage of Swin Transformer. Thus, we designed a unique model. We developed a multi-scale window attention (i.e., the proposed PSRT block) inspired by the classical pyramid structure to provide larger receptive fields and enrich features, thus yielding better information interaction.

E. One Stage in PSRT Block

The use of self-attention in a non-overlapped window is regarded as the fusion of local information. To this end, we designed a burger-like structure for each stage of the PSRT block. One stage in PSRT blocks contains three window self-attention layers, which are shown in Fig. 4-(b). Each stage mainly consists of a Window Multi-head Self-Attention (W-MSA) module, a MultiLayer Perceptron (MLP), and two



(a) One PSRT block consists of three stages (b) One stage in PSRT block

Fig. 4. (a) One PSRT block with three stages, whose window size decreases layer by layer. (b) The components of one stage of the PSRT block.

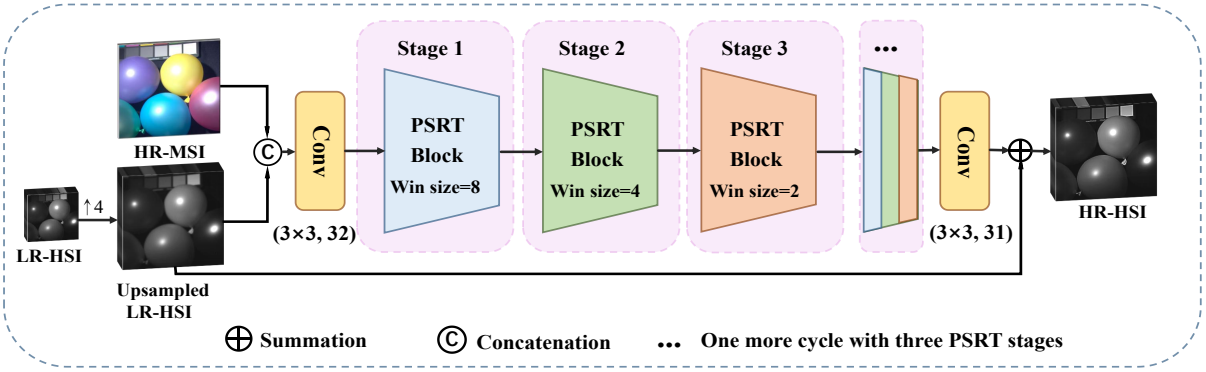


Fig. 5. The overall architecture of the proposed approach using PSRT blocks. "Win size" stands for the size of the window for self-attention.

LayerNorm (LN) layers. The S in the second block stands for Shuffle and the R in the third block indicates that Reshuffle is considered before the W-MSA module. Furthermore, the nonlinearity (i.e., GELU [64]) and the residual connection are exploited in each stage. More specifically, we have:

$$\begin{aligned}
\hat{\mathrm{z}}^{l} &= \text{W-MSA}(\text{LN}(\mathrm{z}^{l-1})) + \mathrm{z}^{l-1}, \\
\mathrm{z}^{l} &= \text{MLP}(\text{LN}(\hat{\mathrm{z}}^{l})) + \hat{\mathrm{z}}^{l}, \\
\hat{\mathrm{z}}^{l+1} &= \text{W-MSA}(\text{LN}(\text{S}(\mathrm{z}^{l}))) + \mathrm{z}^{l}, \\
\mathrm{z}^{l+1} &= \text{MLP}(\text{LN}(\hat{\mathrm{z}}^{l+1})) + \hat{\mathrm{z}}^{l+1}, \\
\hat{\mathrm{z}}^{l+2} &= \text{W-MSA}(\text{LN}(\text{R}(\mathrm{z}^{l+1}))) + \mathrm{z}^{l+1}, \\
\mathrm{z}^{l+2} &= \text{MLP}(\text{LN}(\hat{\mathrm{z}}^{l+2})) + \hat{\mathrm{z}}^{l+2} + \mathrm{z}^{l-1},
\end{aligned} \tag{9}$$

where W-MSA is a window-based self-attention, z^{l-1} is the output in the previous stage, z^{l}, z^{l+1}, and z^{l+2} stand for the outcomes of each stage. We take the "Window self-attention + Shuffle + Window self-attention + Reshuffle + Window self-attention" baseline as one stage of the PSRT block. The self-attention is conducted first on the windows after shuffling, to construct the global dependency among different windows. Then, self-attention is implemented on the windows after reshuffling, to construct the local dependency in one window. By the above-mentioned two strategies of SaR, the information among different windows is propagated.

The overall network architecture is depicted in Fig. 5.

F. Loss Function

L_1 Loss: We calculate the L_1 distance between the network output, I_{MHIF}, and the ground-truth, I_{HR}, in a pixel-wise manner:

$$\mathcal{L}_1 = \|I_{MHIF} - I_{HR}\|_1. \tag{10}$$

SSIM Loss: The Structural SIMilarity (SSIM) can compare the structural differences between I_{MHIF} and I_{HR}, which include the luminance contrast function and the structural contrast function. The SSIM function is defined as:

$$\text{SSIM}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{B}\sum_{i=1}^{B} \frac{(2\mu_{\mathbf{X}^i}\mu_{\hat{\mathbf{X}}^i} + C_1)(2\sigma_{\mathbf{X}^i\hat{\mathbf{X}}^i} + C_2)}{(\mu_{\mathbf{X}^i}^2 + \mu_{\hat{\mathbf{X}}^i}^2 + C_1)(\sigma_{\mathbf{X}^i}^2 + \sigma_{\hat{\mathbf{X}}^i}^2 + C_2)}, \tag{11}$$

where B represents the number of bands, $\mathcal{X}$ and $\hat{\mathcal{X}}$ are equal to $\{\mathbf{X}^i\}_{i=1}^{B}$ and $\{\hat{\mathbf{X}}^i\}_{i=1}^{B}$, respectively, $\mu_{\mathbf{X}^i}$ and $\mu_{\hat{\mathbf{X}}^i}$ are the average of $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$, respectively, $\sigma_{\mathbf{X}^i}^2$ and $\sigma_{\hat{\mathbf{X}}^i}^2$ are the variances of $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$, respectively, $\sigma_{\mathbf{X}^i\hat{\mathbf{X}}^i}$ is the covariance between $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$. C_1 and C_2 are two fixed constants. We use SSIM to measure the image distortion. Thus, the loss is expressed as:

$$\mathcal{L}_{ssim} = 1 - \text{SSIM}(I_{MHIF}, I_{HR}). \tag{12}$$

Overall Loss Function: We optimize the parameters of the network in a unified and end-to-end manner. The overall loss function consists of the weighed sum of two losses:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \lambda_{ssim}\mathcal{L}_{ssim}, \tag{13}$$

where λ_{ssim} is a positive hyperparameter fixed to 0.1 in our experiments.

IV. Experimental Results

This section is devoted to the description of the experimental results to demonstrate the ability of the proposed approach to fuse HSIs and MSIs, getting high-quality HR-HSIs, but even requiring a reduced computational burden. The experimental settings, the benchmarking, and the adopted quality metrics will be described first. Afterwards, the results on four popular datasets will be shown. Finally, an ablation study is provided to the readers.

A. Experimental Settings

Implementation Details: The proposed network is implemented in PyTorch 1.11.0 and Python 3.8.5 using AdamW optimizer with a learning rate of 0.0001 to minimize $\mathcal{L}_{total}$ by 2000 epochs and Windows operating system with NVIDIA GPU GeForce RTX3080. PSRT blocks are initialized using a normal distribution, instead, the initialization of convolution modules is based on a constant distribution.

Datasets: To show the effectiveness of the proposed method, we evaluate the performance on a remote sensing hyperspectral image dataset called Chikusei, which includes agricultural and urban regions in Chikusei, Japan. The image consists of 2517×2335 pixels with 128 spectral bands in the range from 363 nm to 1018 nm. We picked the top-left region with a spatial size of 1000×2200 for training, and we extracted 64×64 overlapping patches from that area as ground-truth. In addition, the sizes of the HR-MSI and LR-HSI patches are $64 \times 64 \times 3$ and $16 \times 16 \times 128$, respectively. We extracted 6 non-overlapping images of $680 \times 680 \times 128$ from the remaining area for testing. The Pavia Centre dataset was collected in 2001 during a flight campaign over the central region of Pavia (Italy) by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. It has 102 spectral bands (the water vapor absorption and noisy spectral bands have been removed from the initial 115 spectral bands) and 1096×1096 pixels in total. We chose a region on the top of the area captured in the dataset as a training set (cropping patches with the same size as the ones in the Chikesei dataset) and the remaining region as a testing set. The testing set consists of four 256×256 non-overlapped hyperspectral patches. To further test the performance of our model, we conducted experiments on the CAVE dataset. CAVE dataset contains 32 HSIs and the corresponding MSIs using red-green-blue (RGB) channels with a size of $512 \times 512 \times 31$. We selected 20 images for training the network, and the remaining 11 images constitute the testing dataset. Instead, Harvard dataset contains 77 HSIs of indoor and outdoor scenes, and each HSI has a size of $1392 \times 1040 \times 31$, covering the spectral range from 420 nm to 720 nm. We cropped the upper left part (1000×1000) of the 20 Harvard images, 10 of which have been used for training, and the rest has been exploited for testing.

Data Simulation: The proposed network requires LR-HSI and HR-MSI $(\mathcal{X}, \mathcal{Y})$ as input pairs and the HR-HSI, $\mathcal{Z}$, as ground-truth (GT) for training. However, the reference HR-HSI, $\mathcal{Z}$, is always unavailable. Hence, a simulation step is needed. Specifically, in the experiments on the CAVE dataset, we cropped the 20 selected training images generating 3920 overlapped patches with size of $64 \times 64 \times 31$ to serve as HR-HSI (ground-truth) $\mathcal{Z}$ patches. Furthermore, we applied a 3×3 Gaussian blur kernel with a standard deviation of 0.5 on the original HR-HSIs to simulate the corresponding LR-HSIs. Then, we downsampled the blurred patches with a scaling factor of 4. Moreover, the HR-MSI patches are generated by the common spectral response function of the Nikon D700 camera. Thus, 3920 RGB image patches with size of $64 \times 64 \times 3$ and LR-HSI patches with size of $16 \times 16 \times 31$ form the input pairs $(\mathcal{X}, \mathcal{Y})$. Afterwards, the pairs and the related GTs have been randomly divided into training data (80%) and validation data (20%). This procedure is similarly applied to the other three datasets to simulate the input LR-HSI and HR-MSI products and the GTs.

B. Benchmarking

To assess the performance of our approach, we compare it with various state-of-the-art methods for MHIF. The upsampled LR-HSI in Fig. 5 is obtained through bicubic interpolation, which is added to the experiments as baseline. Model-based techniques include the CSTF [65] method, the FUSE [66] approach, the GLP-HS [67] method, and the CNN-FUSE [68] approach. In addition, we performed a comparison with other deep learning methods, such as the SSRNet [54], the ResTFNet [56], the MHFNet [57], the HSRNet [30], and the MoG-DCN [58]. All the deep learning approaches are trained with the same input pairs for a fair comparison. Moreover, the related hyperparameters are selected consistently with the original papers.

C. Quality Metrics

Four Quality Indexes (QIs) are adopted to assess the quality of the fusion approaches, including the Peak Signal-to-Noise Ratio (PSNR), the Spectral Angle Mapper (SAM), the Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS), and the SSIM was introduced before.

The PSNR measures the spatial quality of each band in the reconstructed HR-HSI. Thus, we have:

$$\mathrm{PSNR}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{B} \sum_{i=1}^{B} \mathrm{PSNR}(\mathbf{X}^{\mathrm{i}}, \hat{\mathbf{X}}^{\mathrm{i}}), \tag{14}$$

where $\mathbf{X}^i \in \mathbb{R}^{H \times W}$ and $\hat{\mathbf{X}}^i \in \mathbb{R}^{H \times W}$ represent the i-th band of the $\mathcal{X} \in \mathbb{R}^{H \times W \times B}$ and $\hat{\mathcal{X}} \in \mathbb{R}^{H \times W \times B}$, respectively, and $\mathrm{PSNR}(\cdot, \cdot)$ is the PSNR function defined as:

$$\mathrm{PSNR}(\mathbf{X}^i, \hat{\mathbf{X}}^i) = 20 \cdot \log_{10} \left(\frac{\max\left(\mathbf{X}^i\right)}{\sqrt{\mathrm{MSE}(\mathbf{X}^i, \hat{\mathbf{X}}^i)}} \right), \tag{15}$$

where $\mathrm{MSE}(\mathbf{X}^i, \hat{\mathbf{X}}^i)$ is the mean square error between $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$ and $\max(\cdot)$ is the maximum operator applied to an image $\mathbf{X}^i$.

The SAM measures the spectral distortion of each hyperspectral pixel in the reconstructed HR-HSI. Thus, we have:

$$\mathrm{SAM}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{HW} \sum_{i=1}^{HW} \cos^{-1} \left(\frac{\mathbf{x}_i^T \hat{\mathbf{x}}_i}{\|\mathbf{x}_i\|_2 \|\hat{\mathbf{x}}_i\|_2} \right), \quad (16)$$

where $\cos^{-1}$ is the arccosine function, $\mathbf{x}_i \in \mathbb{R}^{B\times 1}$ and $\hat{\mathbf{x}}_i \in \mathbb{R}^{B\times 1}$ denote the spectra of the i-pixel of $\mathcal{X}$ and $\hat{\mathcal{X}}$, respectively, $\|\cdot\|_2$ is the ℓ_2 norm, and $\cdot^T$ denotes the transpose operator.

The ERGAS considers the ratio of the ground sample distances between HR-MSI and LR-HSI to measure the global statistical quality of the reconstructed HR-HSI. The ERGAS is formulated as:

$$\mathrm{ERGAS}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{100}{c} \sqrt{\frac{1}{B} \sum_{i=1}^{B} \frac{\mathrm{MSE}(\mathbf{X}^i, \hat{\mathbf{X}}^i)}{\mu_{\hat{\mathbf{X}}^i}^2}}, \quad (17)$$

where c denotes the scaling factor and $\mu_{\hat{\mathbf{X}}^i}^2$ is the square of the mean value of $\hat{\mathbf{X}}^i$.

The SSIM is introduced in the section related to the description of the adopted loss function. The higher the PSNR value, the better the performance. In contrast, smaller values of SAM and ERGAS indicate better quality of the reconstructed HR-HSI. SSIM ranges from -1 to 1, and values closer to 1 indicate a better quality of the fused product. Finally, optimal values are $+\infty$ for PSNR, 0 for SAM and ERGAS, and 1 for SSIM.

D. Results on Chikusei Dataset

We conducted experiments to evaluate our network on real remote-sensing images. We treated the original data as ground-truth and simulated the LR-HSI in the same way as described in Sect. IV-A. The HR-HSI is obtained by the hyperspectral camera and the corresponding HR-MSI is captured by the Canon EOS 5D Mark II. In Fig. 6, the 6 non-overlapping testing images have been presented. All the deep learning-based methods are re-trained on training data extracted from the Chikusei dataset.

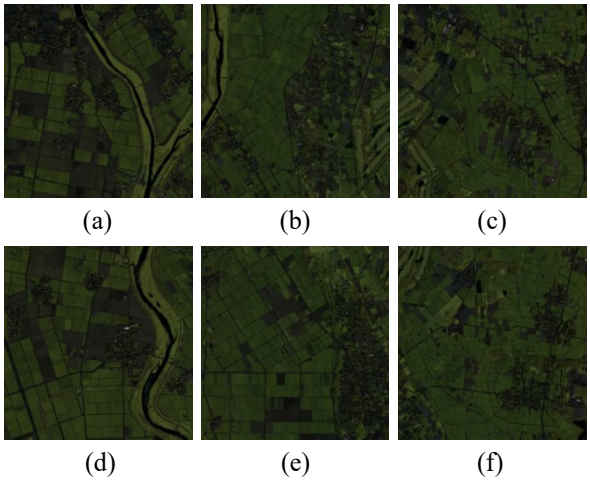


Fig. 6. The 6 testing images from the Chikusei dataset: (a) *area 1*, (b) *area 2*, (c) *area 3*, (d) *area 4*, (e) *area 5*, (f) *area 6*. A pseudo-color representation is used combining the 119th, the 90th, and the 69th bands.

TABLE I
THE FOUR AVERAGE QIS AND THE CORRESPONDING PARAMETERS ON THE 6 TESTING IMAGES FROM THE CHIKUSEI DATASET. THE BEST VALUES ARE HIGHLIGHTED IN BOLD, THE SECOND BEST VALUES ARE UNDERLINED. M MEANS MILLIONS.

Method	PSNR	SAM	ERGAS	SSIM	# params
Bicubic	30.16	2.98	6.67	0.912	/
GLP-HS [67]	31.60	3.29	5.69	0.920	/
FUSE [66]	27.76	4.80	7.23	0.882	/
CSTF [65]	30.36	4.58	5.91	0.824	/
CNN-FUSE [68]	31.83	4.76	5.25	0.918	/
SSRNet [54]	35.54	2.33	3.79	<u>0.954</u>	**0.03M**
ResTFNet [56]	36.70	2.20	3.66	0.949	2.26M
MHFNet [57]	33.19	3.18	6.24	0.927	3.63M
HSRNet [30]	**36.95**	2.08	3.60	0.952	1.90M
MoG-DCN [58]	36.04	<u>2.04</u>	<u>3.55</u>	0.949	53.19M
Our	<u>36.83</u>	**2.01**	**3.54**	**0.955**	<u>1.05M</u>

The average four QIs and the corresponding parameters for the 6 testing images of the Chikusei dataset are shown in Tab. I. Compared with the traditional methods, the deep learning-based approaches obtain better performance due to the inductive bias ability. For instance, the deep neural network structure is based on the principle that the hierarchical processing of information can get better results. Instead, CNNs made the hypothesis that the information has a spatial locality and they used sliding convolution to share weights to reduce the parameter space. Finally, Transformer can establish long-distance dependence. More specifically, our method achieves state-of-the-art results compared with the benchmarking for most of the quality metrics. Regarding to the comparison with the MoG-DCN [58] (the second best approach), our method gets a better PSNR (higher than 0.79 dB) and lower SAM and ERGAS values also requiring fewer parameters. Compared with the third best method, HSRNet [30], it is clear that our approach outperforms it considering 3 out of 4 QIs, i.e. SSIM, SAM, and ERGAS. In terms of qualitative assessment, see Fig. 7, we present the pseudo-color representations of the fused products, and some error maps to aid the visual inspection. Compared with the benchmark, our approach has better detail recovery and visual effects. More specifically, FUSE [66] shows color changes and blurred effects, see also Tab. I. Having a look at the error maps, the reconstruction provided by our model is close to the ground-truth and surely closer than the compared approaches. Thus, the proposed approach achieves the best results in terms of image detail reconstruction and the darkest colors in the related error map.

E. Results on Pavia Centre Dataset

In this section, we assess the performance of another real remote sensing dataset (i.e., Pavia Centre). We considered the original HSI as ground-truth, and we simulated the LR-HSI in the same way as in Sect. IV-A.

In Fig. 8, the 4 testing images have been presented. We can see from Fig. 9 that the proposed approach gets high performance. In particular, the residual maps demonstrate that there is little difference between the result from our approach

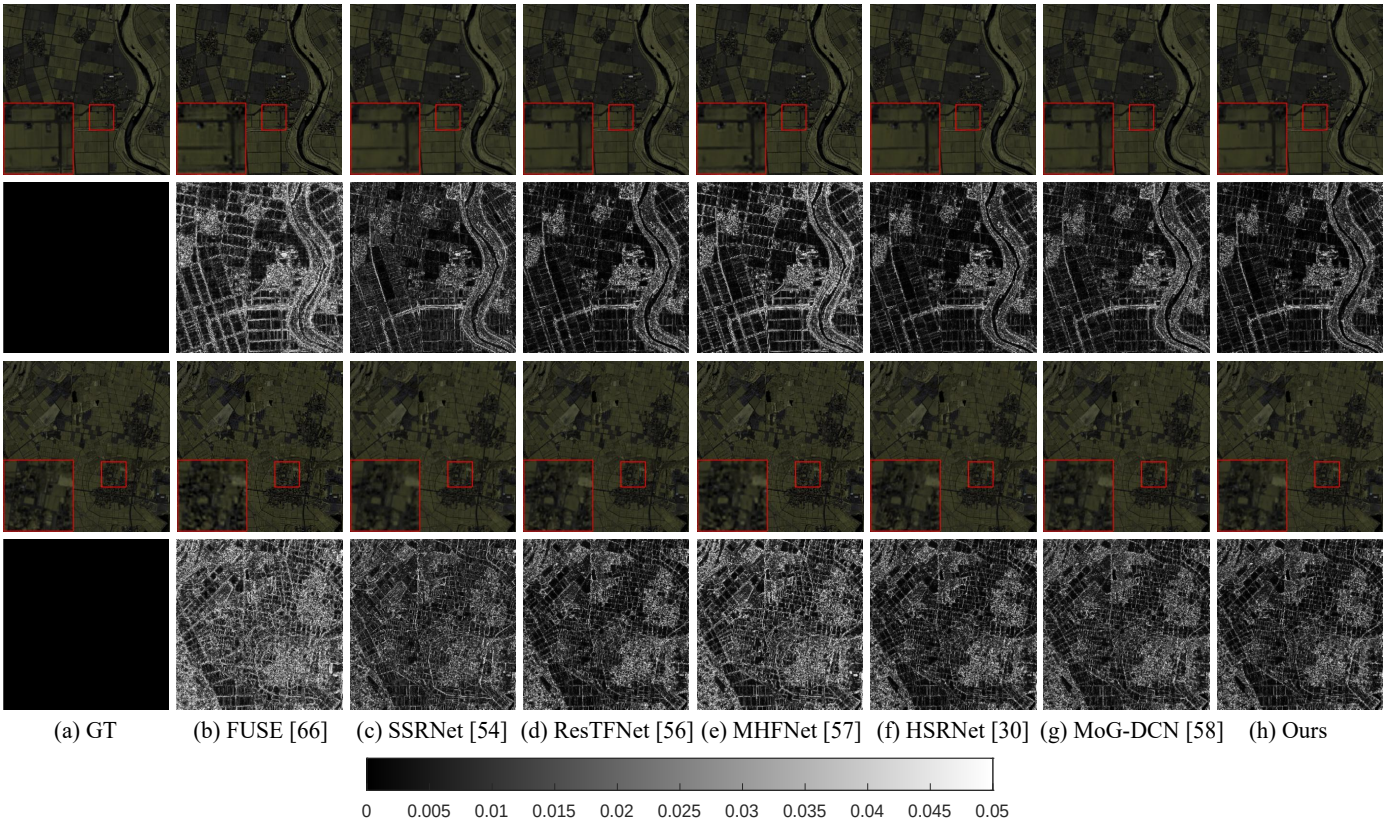


Fig. 7. The first and third rows show the results using the pseudo-color representation on "*area 4*" and "*area 6*", respectively, from the Chikusei dataset. Some close-ups are depicted in the red rectangles. The second and fourth rows show the residuals between the GT and the fused products. (a) GT, (b) FUSE [66], (c) SSRNet [54], (d) ResTFNet [56], (e) MHFNet [57], (f) HSRNet [30], (g) MoG-DCN [58], and (h) Ours.

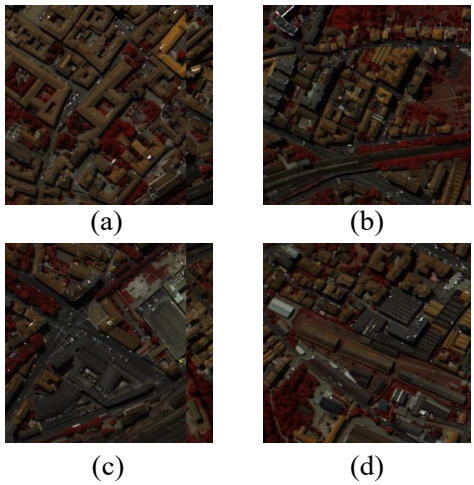


Fig. 8. The 4 testing images from the Pavia Centre dataset: (a) *area 1*, (b) *area 2*, (c) *area 3*, (d) *area 4*. A pseudo-color representation is used combining the 64th, the 32nd, and the 11th bands.

and the GT. A numerical assessment is reported in Tab. II. It can be found that deep learning approaches outperform traditional ones. Our method gets overall good results (in agreement with high-performance approaches such as HSRNet and MoG-DCN). Moreover, it represents an efficient solution supported by a lightweight architecture with a reduced number of parameters. To sum up, the proposed approach shows an outstanding trade-off between performance and computational costs on the Pavia Centre dataset.

TABLE II
THE FOUR AVERAGE QIS AND THE CORRESPONDING PARAMETERS ON THE 4 TESTING IMAGES FROM THE PAVIA CENTRE DATASET. THE BEST VALUES ARE HIGHLIGHTED IN BOLD, THE SECOND BEST VALUES ARE UNDERLINED. M MEANS MILLIONS.

Method	PSNR	SAM	ERGAS	SSIM	# params
Bicubic	19.00	6.45	9.64	0.573	/
GLP-HS [67]	30.74	4.82	5.55	0.897	/
FUSE [66]	36.66	6.18	3.53	0.953	/
CSTF [65]	45.24	2.22	1.14	0.993	/
CNN-FUSE [68]	43.88	2.68	1.36	0.989	/
SSRNet [54]	46.50	1.73	1.00	0.996	**0.22M**
ResTFNet [56]	41.07	2.21	1.45	0.993	2.44M
MHFNet [57]	35.42	3.98	3.84	0.946	1.90M
HSRNet [30]	45.56	1.66	<u>1.00</u>	0.995	3.10M
MoG-DCN [58]	<u>48.24</u>	**1.48**	**0.86**	**0.997**	7.69M
Our	**48.26**	<u>1.53</u>	1.24	<u>0.996</u>	<u>0.59M</u>

F. Results on CAVE Dataset

We also tested our model on the CAVE dataset. Fig. 10 presents the 11 testing images in an RGB color composition.

From Tab. III, we can see that the proposed approach (with only $0.25M$ parameters) overcomes the other methods in 3 out of 4 QIs, i.e. PSNR, SAM, and SSIM. Specifically, we observed an improvement of $\sim$ 0.99/5.72/0.06% in PSNR/SAM/SSIM when compared with the second best

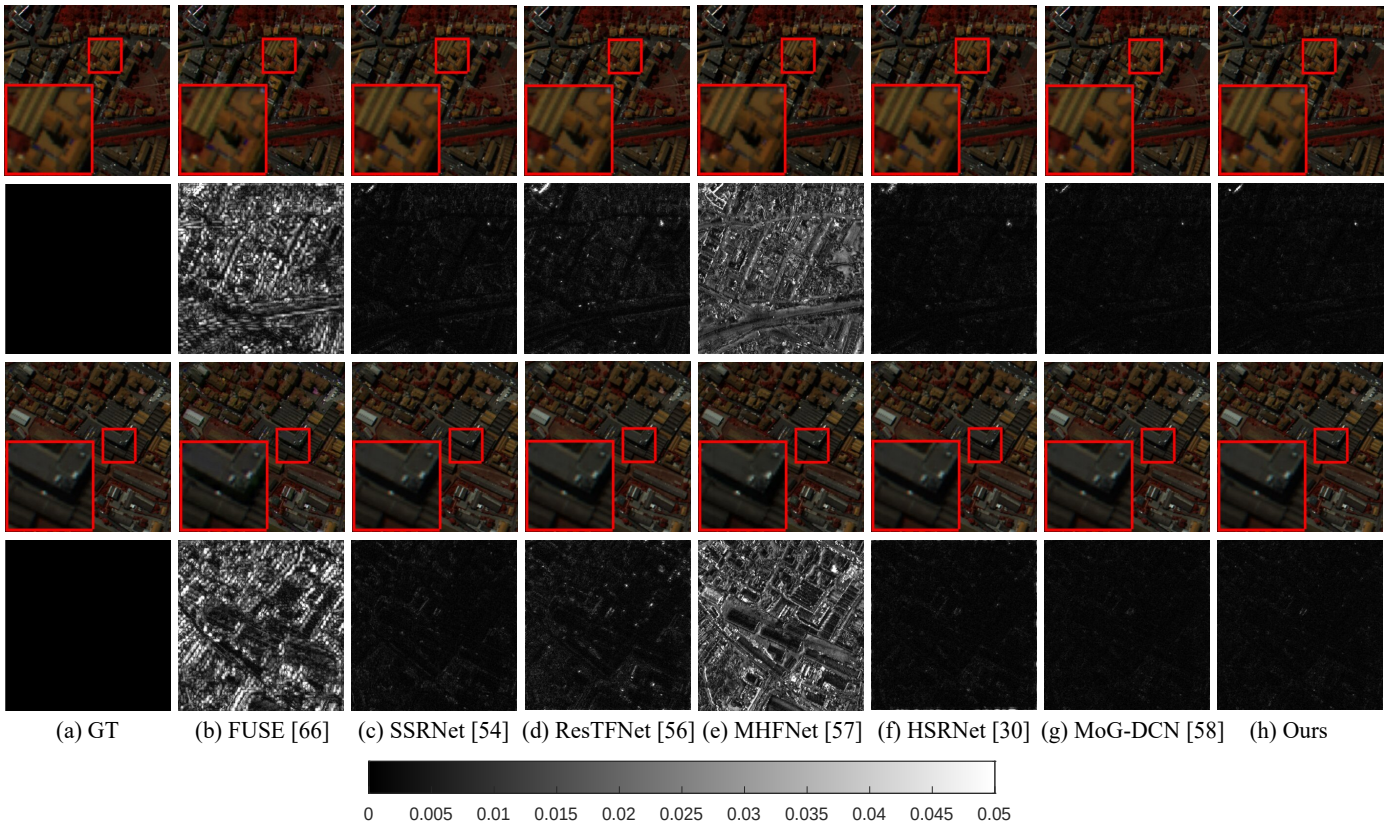


Fig. 9. The first and third rows show the results using the pseudo-color representation on "*area 2*" and "*area 4*", respectively, from the Pavia Centre dataset. Some close-ups are depicted in the red rectangles. The second and fourth rows show the residuals between the GT and the fused products. (a) GT, (b) FUSE [66], (c) SSRNet [54], (d) ResTFNet [56], (e) MHFNet [57], (f) HSRNet [30], (g) MoG-DCN [58], and (h) Ours.

Fig. 10. The testing images from the CAVE dataset: (a) *balloons*, (b) *cd*, (c) *chart and stuffed toy*, (d) *clay*, (e) *fake and real beers*, (f) *fake and real lemon slices*, (g) *fake and real tomatoes*, (h) *feathers*, (i) *flowers*, (j) *hairs*, and (k) *jelly beans*. An RGB color representation is used to depict the images.

TABLE III
THE FOUR AVERAGE QIS AND THE CORRESPONDING PARAMETERS ON THE 11 TESTING IMAGES FROM THE CAVE DATASET. M MEANS MILLIONS.

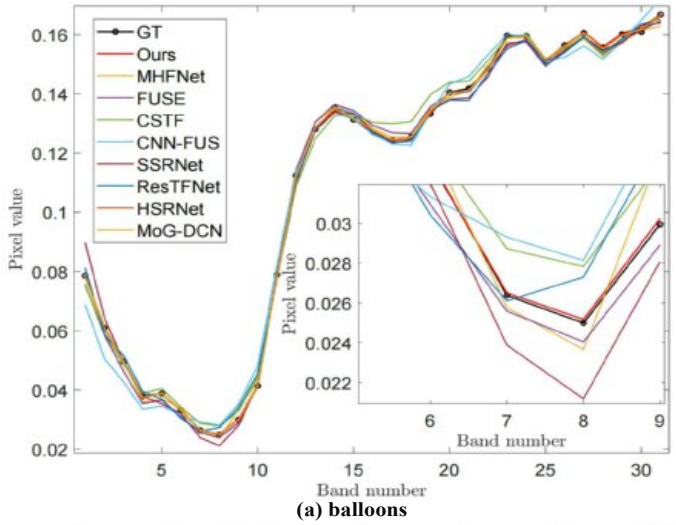
Method	PSNR	SAM	ERGAS	SSIM	# params
Bicubic	32.86	4.28	7.19	0.945	/
GLP-HS [67]	37.81	5.36	4.66	0.972	/
FUSE [66]	39.72	5.83	4.18	0.975	/
CSTF [65]	42.14	9.92	3.07	0.964	/
CNN-FUSE [68]	42.66	6.44	2.95	0.982	/
SSRNet [54]	45.28	4.72	2.06	0.990	**0.03M**
ResTFNet [56]	45.35	3.76	1.98	0.992	2.26M
MHFNet [57]	46.32	4.33	1.74	0.992	3.63M
HSRNet [30]	47.82	2.66	**1.34**	0.995	1.90M
MoG-DCN [58]	48.30	2.62	1.36	0.995	47.28M
Our	**48.78**	**2.47**	1.66	**0.995**	0.25M

method, i.e., MoG-DCN [58]. Compared with the third-best method, HSRNet [30], our approach improves $\sim$ 2.01/7.14/0.07% in PSNR/SAM/SSIM with fewer parameters.

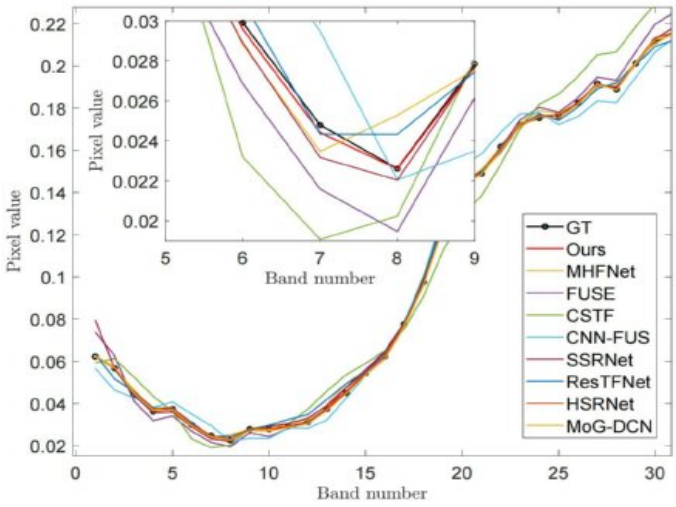
By comparing the results in Fig. 12 and the related close-ups in the rectangular boxes, we can remark the best performance of the proposed approach. The colors of the objects in the figures, the shape of the details, and the edges are closer to the GT. On the other hand, the residual maps show that the gap between our outcome and the GT is minimal. Finally, we illustrated possible spectral distortions in the fused products by showing spectral vectors. Fig. 11 depicts the spectral vectors for the 31 bands at position (386, 61). For convenience, we zoomed in the spectral vectors of five bands (5th$\sim$9th bands), see the rectangular boxes in Fig. 11. It can be seen in both the figures that the spectral vectors of the proposed method (the red lines) are the closest to the GT.

G. Results on Harvard Dataset

We conducted experiments on the Harvard dataset to assess the performance of our network. Again, we used the same simulation approach as in Sect. IV-A. The 10 testing images, randomly selected from the Harvard dataset, are depicted in Fig. 13. Tab. IV reports the results.



(a) balloons



(b) chart and stuffed toy

Fig. 11. Spectral vectors of the GT and the benchmark. (a) Spectral vectors in "*balloons*" located at position (386, 61). (b) Spectral vectors in "*chart and stuffed toy*" located at position (386, 61).

TABLE IV
THE FOUR AVERAGE QIS AND THE CORRESPONDING PARAMETERS ON 10 TESTING IMAGES FROM THE HARVARD DATASET. M MEANS MILLIONS.

Method	PSNR	SAM	ERGAS	SSIM	# params
Bicubic	35.01	2.83	4.58	0.945	/
GLP-HS [67]	40.14	3.52	3.74	0.966	/
FUSE [66]	42.06	3.23	3.14	0.977	/
CSTF [65]	43.04	3.29	2.39	0.972	/
CNN-FUS [68]	43.61	3.32	2.78	0.978	/
SSRNet [54]	44.40	2.61	2.39	0.985	**0.03M**
ResTFNet [56]	44.47	2.56	2.21	0.985	2.26M
MHFNet [57]	43.10	2.76	3.28	0.977	3.63M
HSRNet [30]	44.29	2.66	2.45	0.984	1.90M
MoG-DCN [58]	45.82	**2.22**	**1.99**	**0.987**	47.28M
Our	**47.02**	2.35	2.13	0.986	0.25M

Because of the presence of high-frequency noise in the images, the results recovered by each model have both spatial and spectral distortions. Our model achieves the best performance on the PSNR metric and the second-least number of parameters compared with state-of-the-art methods. Specifically, comparing it with the second-best method, i.e. MoG-DCN [58], it is straightforward that our method outperforms it by 1.2 dB in the PSNR index with a number of parameters less than 189 times the ones of MoG-DCN. Having a look at the results of HSRNet [30], our approach has better QIs. In particular, a higher PSNR (2.73 dB more) and SSIM, and lower values of SAM and ERGAS with fewer parameters. From a qualitative point of view, in Fig. 14, our model still has a great visual rendering showing an excellent details reconstruction.

H. Ablation Study

This section is devoted to the presentation of the results about the ablation study to assess the performance of the multi-scale windows and the SaR strategy in the PSRT. For the sake of brevity and without affecting the generality, the analysis is provided considering the CAVE dataset.

1) Shuffle and Reshuffle: Ablation studies on the SaR strategy are reported in Tab. V. $w/o\ SaR\_1$ indicates that we removed the SaR strategy at the first stage, and $w/o\ SaR\_2$ denotes that we removed the SaR strategy at the first two stages. Instead, $w/o\ SaR\_3$ means that the whole PSRT blocks do not use the SaR strategy. It is clear that removing the SaR strategy at a stage leads to a decrease in performance. Moreover, to test the performance of the Shifted Window approach, we replace the SaR strategy with this latter (called "shifted" in Tab. V). We can note that with this configuration, we have a clear decrease in performance.

TABLE V
THE AVERAGE FOUR QIS VARYING THE APPLICATION OF THE SAR APPROACH ON THE CAVE DATASET. W/O SAR_x INDICATES THAT THE SAR STRATEGY IS REMOVED FROM THE FIRST STAGE TO THE x-TH STAGE. SHIFTED MEANS THAT WE SUBSTITUTE OUR SAR STRATEGY WITH THE SHIFTED WINDOW APPROACH [1].

Method	PSNR	SAM	ERGAS	SSIM
w/o SaR_1	48.42	2.61	1.95	0.995
w/o SaR_2	48.36	2.60	1.90	0.995
w/o SaR_3	48.35	2.55	2.18	0.995
shifted	48.20	2.58	2.35	0.995
SaR	**48.78**	**2.47**	**1.66**	**0.995**

2) Multi-Scale Windows: Tab. VI reports the results of the PSRT blocks with all the windows fixed to the same size. It is easy to note a degradation in performance if each PSRT block uses a fixed-size window. We tested three different configurations with size of 2, 4, and 8. The larger the size of the window, the better the performance. Anyway, this test proves the necessity of a multi-scale analysis requiring different window sizes at different stages.

3) SaR Strategy on Swin Transformer: To further demonstrate the effectiveness of the SaR strategy, we tested it against the Shifted Window operation. We designed Swin Transformer structures with three different sizes comparing the Shifted Window approach (Swin-Shift) with our SaR strategy (Swin-SaR). The horizontal axis in Fig. 15 represents the parameter amount and the vertical axis is related to the PSNR. The purple dots represent the Swin Transformer with the Shifted Window

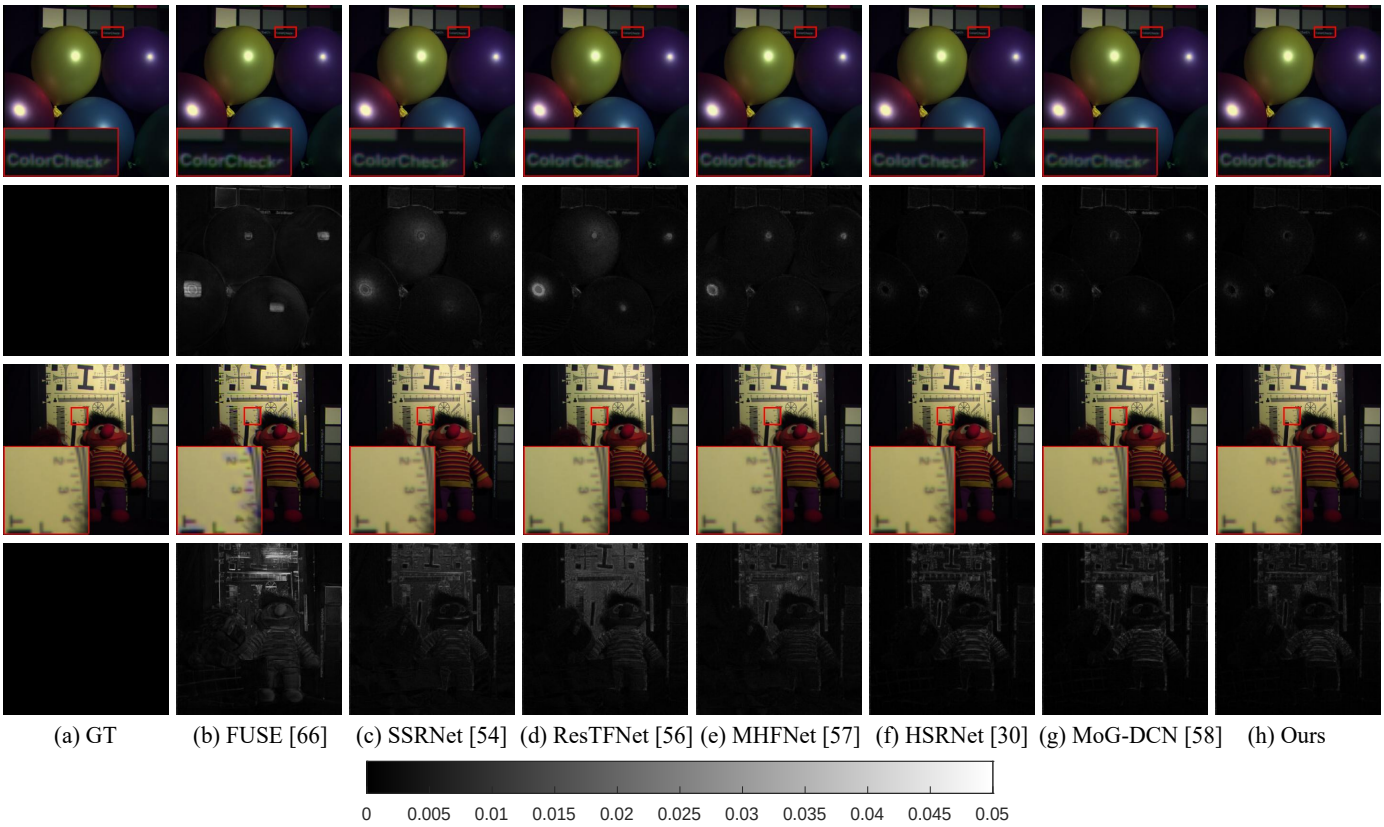


Fig. 12. The first and third rows show the results using the true color representation on "*balloons*" and "*chart and stuffed toy*", respectively, from the CAVE dataset. Some close-ups are depicted in the red rectangles. The second and fourth rows show the residuals between the GT and the fused products. (a) GT, (b) FUSE [66], (c) SSRNet [54], (d) ResTFNet [56], (e) MHFNet [57], (f) HSRNet [30], (g) MoG-DCN [58], and (h) Ours.

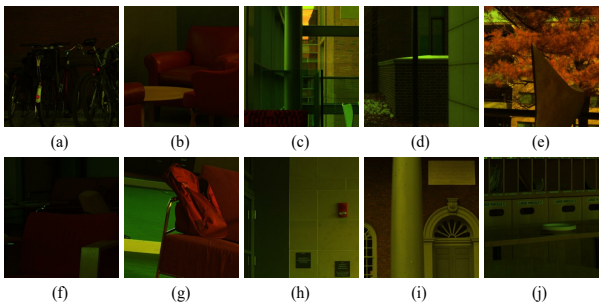


Fig. 13. The 10 testing images from the Harvard dataset: (a) *bikes*, (b) *sofa1*, (c) *window*, (d) *fence*, (e) *tree*, (f) *sofa2*, (g) *backpack*, (h) *wall*, (i) *door*, and (j) *parcels*. A pseudo-color representation is used combining the 30th, the 15th, and the 2nd bands

TABLE VI
THE AVERAGE FOUR QIS VARYING THE SIZE OF THE WINDOWS ON THE CAVE DATASET. WE INDICATE WITH SIZE=x THAT THE SIZE OF THE WINDOW IS FIXED TO x.

Method	Backbone	PSNR	SAM	ERGAS	SSIM
win size=2	PSRT	47.74	2.66	2.65	0.995
win size=4	PSRT	48.16	2.64	2.20	0.995
win size=8	PSRT	<u>48.42</u>	<u>2.57</u>	<u>1.99</u>	<u>0.995</u>
Pyramid	PSRT	**48.78**	**2.47**	**1.66**	**0.995**

approach. Instead, the blue dots indicate the Swin Transformer with the proposed SaR strategy. It can be noted that the positive effects of Swin Transformer are improved when the Shifted Window operation is replaced by our SaR strategy. This proves that the proposed SaR strategy is better than the Shifted Window approach when applied to Swin Transformer for solving the MHIF task.

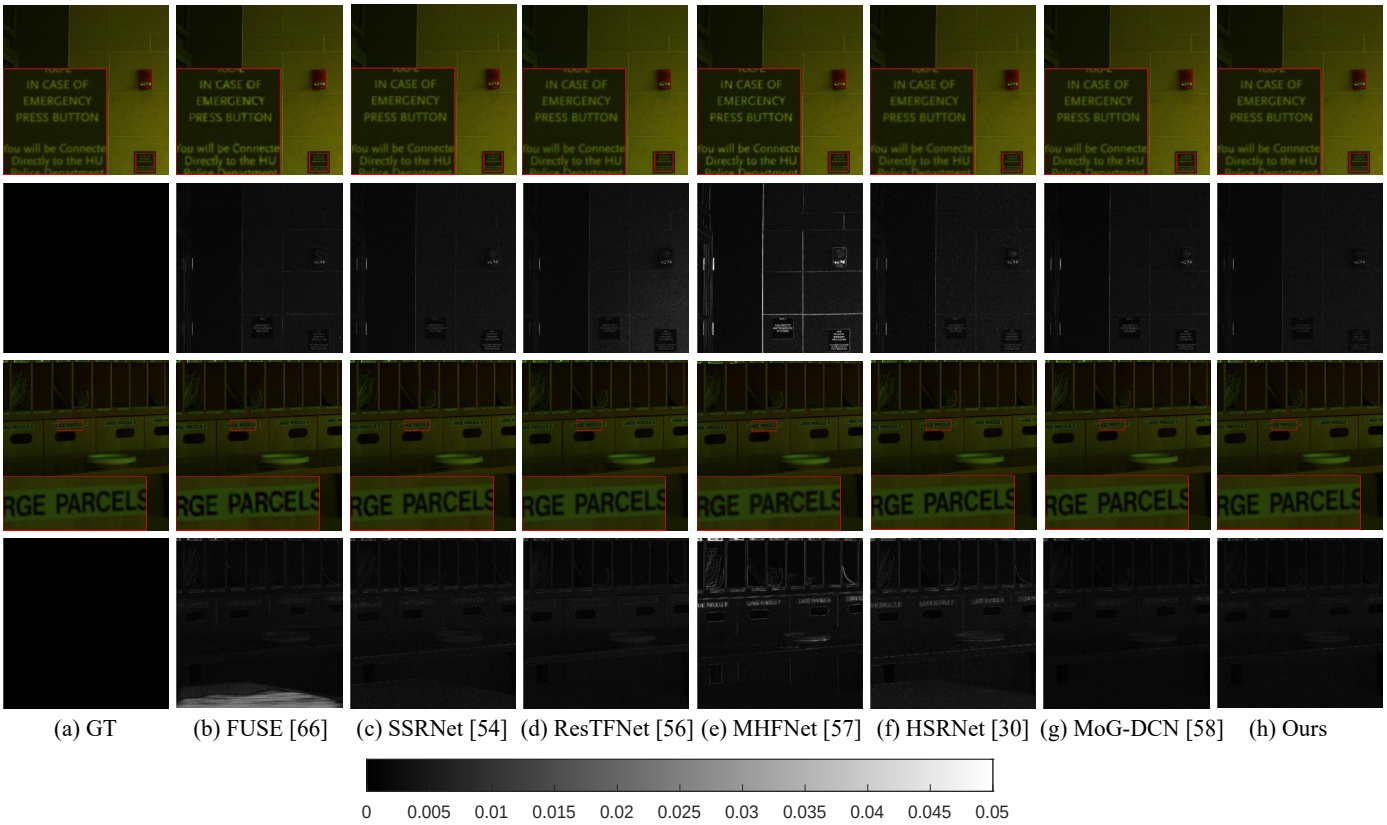
4) SaR Strategy with another shuffle strategy: We considered the following experiments to confirm the superiority of our SaR approach. We employed PSRT as the backbone, replacing our SaR strategy with the Shuffle Transformer approach in [62] and the OCnet method in [63]. In Tab. VII, we can easily observe that the SaR strategy outperforms the above-mentioned methods.

TABLE VII
THE AVERAGE FOUR QIS REPLACING THE SAR STRATEGY WITH THE SHUFFLE TRANSFORMER APPROACH [62] (SHUFFLE) AND WITH THE OCNET METHOD [63] (OC) ON THE CAVE DATASET.

Method	PSNR	SAM	ERGAS	SSIM
Shuffle	47.07	3.13	<u>2.60</u>	<u>0.994</u>
OC	48.56	**2.46**	2.66	**0.995**
SaR	**48.78**	<u>2.47</u>	**1.66**	**0.995**

I. *Experiments with Scaling Factor of 8*

Focusing again on the CAVE dataset, we assessed the performance of the compared approaches simulating a scaling factor of 8. The results are reported in Tab. VIII. The fusion results show that our method still achieves the best results with a small amount of parameters.



(a) GT (b) FUSE [66] (c) SSRNet [54] (d) ResTFNet [56] (e) MHFNet [57] (f) HSRNet [30] (g) MoG-DCN [58] (h) Ours

Fig. 14. The first and third rows show the results using the pseudo-color representation on "*wall*" and "*parcels*", respectively, from the Harvard dataset. Some close-ups are depicted in the red rectangles. The second and fourth rows show the residuals between the GT and the fused products. (a) GT, (b) FUSE [66], (c) SSRNet [54], (d) ResTFNet [56], (e) MHFNet [57], (f) HSRNet [30], (g) MoG-DCN [58], and (h) Ours.

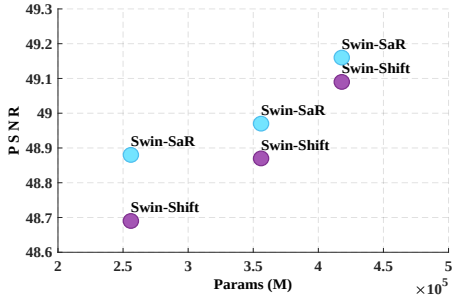


Fig. 15. PSNR Vs. Parameter amount for Swin-Shift and Swin-SaR on the CAVE dataset.

TABLE VIII
THE AVERAGE FOUR QIS AND THE CORRESPONDING PARAMETERS ON THE CAVE DATASET SIMULATING A SCALING FACTOR OF 8. M MEANS MILLIONS.

Method	PSNR	SAM	ERGAS	SSIM	# params
Bicubic	28.42	5.62	11.09	0.890	/
FUSE [66]	36.18	9.87	1.57	0.927	/
CSTF [65]	39.13	15.61	1.19	0.946	/
CNN-FUS [68]	38.20	9.57	2.32	0.955	/
SSRNet [54]	43.79	5.09	1.23	0.988	**0.03M**
ResTFNet [56]	43.21	4.69	1.32	0.990	2.26M
MHFNet [57]	45.00	4.88	0.99	0.990	3.63M
HSRNet [30]	44.97	3.33	0.94	0.992	1.90M
MoG-DCN [58]	46.08	3.33	**0.90**	0.993	47.28M
Our	**46.09**	**3.04**	1.73	**0.993**	0.25M

J. Image Boundary Influence

Having a look at the Swin Transformer, it can be easily noted that windows on image boundaries shrink producing poor results. Thus, we tested the performance on image boundaries using the PSRT as backbone to compare the SaR strategy with the Shifted window approach [1]. We cropped regions with strides of 4, 8, and 12 pixels from image edges. It can be seen from Tab. IX that our SaR strategy outperforms the Shifted window approach on image boundaries.

V. CONCLUSIONS

In this paper, we proposed an approach to take into account of the modeling of global information while reducing computation complexity to solve the MHIF task. Specifically, the SaR strategy applied along the horizontal and vertical directions is adopted to get better window interactions. One stage of the PSRT is the combination of "Window self-attention + Shuffle + Window self-attention + Reshuffle + Window self-attention", which is similar to a burger-like structure. SaR strategy with window self-attention allows spreading the local information without adding extra computation. Meanwhile, the design relied upon the concept of multi-scale analysis using pyramidal structures enables the model to extract information at different resolutions; an approach that is more effective for the MHIF task. Extensive experiments demonstrated the advantages of each module of the proposed technique over-

TABLE IX
RESULTS ABOUT THE IMAGE BOUNDARY INFLUENCE. THE AVERAGE FOUR QIS ARE CALCULATED ON THE CAVE DATASET WITH A SCALING FACTOR OF 4. SAR_x INDICATES THE RESULTS OF THE PSRT APPLYING THE SAR STRATEGY ON BOUNDARY REGIONS OF WIDTH OF x PIXELS. SHIFTED_x DENOTES THE RESULTS OF THE PSRT APPLYING THE SHIFTED WINDOW APPROACH ON BOUNDARY REGIONS OF WIDTH OF x PIXELS. THE BEST VALUES ARE HIGHLIGHTED IN BOLD, THE SECOND BEST VALUES ARE UNDERLINED.

Method	Backbone	PSNR	SAM	ERGAS	SSIM
Shifted_4	PSRT	35.73	16.20	5.84	0.972
SaR_4	PSRT	36.09	16.33	5.91	0.972
Shifted_8	PSRT	41.05	9.93	3.77	0.986
SaR_8	PSRT	41.44	9.84	3.74	0.986
Shifted_12	PSRT	<u>44.07</u>	<u>7.45</u>	**2.88**	<u>0.990</u>
SaR_12	PSRT	**44.10**	**7.45**	<u>2.90</u>	**0.990**

coming the performance of several state-of-the-art methods for MHIF.

REFERENCES

[1] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 012–10 022.

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

[3] W. Wang, X. Fu, W. Zeng, L. Sun, R. Zhan, Y. Huang, and X. Ding, "Enhanced deep blind hyperspectral image fusion," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[4] M. Zhou, X. Fu, J. Huang, F. Zhao, A. Liu, and R. Wang, "Effective pan-sharpening with transformer and invertible neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2021.

[5] X. Cao, X. Fu, D. Hong, Z. Xu, and D. Meng, "Pancsc-net: A model-driven deep unfolding method for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2021.

[6] J. Liu, Z. Wu, L. Xiao, and X. Wu, "Model inspired autoencoder for unsupervised hyperspectral image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[7] W. Sun, K. Ren, X. Meng, C. Xiao, G. Yang, and J. Peng, "A band divide-and-conquer multispectral and hyperspectral image fusion method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[8] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 60–88, 2020.

[9] J. Hu, T. Huang, L. Deng, H. Dou, D. Hong, and G. Vivone, "Fusformer: A transformer-based fusion network for hyperspectral image super-resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[10] Y. Mo, X. Kang, P. Duan, and S. Li, "A robust uav hyperspectral image stitching method based on deep feature matching," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[11] R. Dian, S. Li, B. Sun, and A. Guo, "Recent advances and new guidelines on hyperspectral and multispectral image fusion," *Information Fusion*, vol. 69, pp. 40–51, 2021.

[12] R. Dian, L. Fang, and S. Li, "Hyperspectral image super-resolution via non-local sparse tensor factorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5345–5355, 2018.

[14] G. Vivone, M. Dalla Mura, A. Garzelli, R. Restaino, G. Scarpa, M. O. Ulfarsson, L. Alparone, and J. 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*, vol. 9, no. 1, pp. 53–81, 2020.

[15] G. Vivone, P. Addesso, and J. Chanussot, "A combiner-based full resolution quality assessment index for pansharpening," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 3, pp. 437–441, 2018.

[16] R. Dian, S. Li, L. Fang, T. Lu, and J. M. Bioucas-Dias, "Nonlocal sparse tensor factorization for semiblind hyperspectral and multispectral image fusion," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4469–4480, 2020.

[17] Y. Zhuo, T. Zhang, J. Hu, H. Dou, T. Huang, and L. Deng, "A deep-shallow fusion network with multidetail extractor and spectral attention for hyperspectral pansharpening," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 7539–7555, 2022.

[18] J. Hu, T. Huang, L. Deng, H. Dou, and G. Hong, Danfeng adn Vivone, "Fusformer: A transformer-based fusion network for hyperspectral image super-resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[19] T. Xu, T. Huang, L. Deng, and N. Yokoya, "An iterative regularization method based on tensor subspace representation for hyperspectral image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[20] M. Uzair, A. Mahmood, and A. S. Mian, "Hyperspectral face recognition using 3d-dct and partial least squares." in *Proceedings of theThe British Machine Vision Conference (BMVC)*, vol. 1, 2013, p. 10.

[21] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," in *Proceedings of the IEEE*, vol. 101, no. 3. IEEE, 2012, pp. 652–675.

[22] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2021.

[23] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5384–5394, 2019.

[24] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[25] R. Hang, Z. Li, P. Ghamisi, D. Hong, G. Xia, and Q. Liu, "Classification of hyperspectral and lidar data using coupled cnns," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4939–4950, 2020.

[26] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 5, pp. 1267–1279, 2009.

[27] X. Tai, L. Deng, and K. Yin, "A multigrid algorithm for maxflow and min-cut problems with applications to multiphase image segmentation," *Journal of Scientific Computing*, vol. 87, no. 101, 2021.

[28] T. Xu, T. Huang, L. Deng, X. Zhao, and J. Huang, "Hyperspectral image superresolution using unidirectional total variation with tucker decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4381–4398, 2020.

[29] Z. Jin, L. Deng, T. Zhang, and X. Jin, "Bam: Bilateral activation mechanism for image fusion," in *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2021, pp. 4315–4323.

[30] J. Hu, T. Huang, L. Deng, T. Jiang, G. Vivone, and J. Chanussot, "Hyperspectral image super-resolution via deep spatiospectral attention convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, p. 15, 2021.

[31] J. Xiao, T. Huang, L. Deng, Z. Wu, and G. Vivone, "A new context-aware details injection fidelity with adaptive coefficients estimation for variational pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[32] C. Jin, L. Deng, T. Huang, and G. Vivone, "Laplacian pyramid networks: A new approach for multispectral pansharpening," *Information Fusion*, vol. 78, pp. 158–170, 2022.

[33] Z. Wu, T. Huang, L. Deng, J. Hu, and G. Vivone, "Vo+net: An adaptive approach using variational optimization and deep learning for panchromatic sharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[34] J. Huang, T. Huang, X. Zhao, and L. Deng, "Nonlocal tensor-based sparse hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6854–6868, 2021.

[35] Z. Wu, T. Huang, L. Deng, G. Vivone, J. Miao, J. Hu, and X. Zhao, "A new variational approach based on proximal deep injection and gradient intensity similarity for spatio-spectral image fusion," *IEEE Journal of*

Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 6277–6290, 2020.

[36] L. Deng, R. Glowinski, and X. Tai, "A new operator splitting method for the euler elastica model for image smoothing," *SIAM Journal on Imaging Sciences*, vol. 12, no. 2, pp. 1190–1230, 2019.

[37] L. Deng, G. Vivone, M. E. Paoletti, G. Scarpa, J. He, Y. Zhang, C. Jocelyn, and P. Antonio, "Machine learning in pansharpening: A benchmark, from shallow to deep networks," *IEEE Geoscience and Remote Sensing Magazine*, pp. 2–38, 2022.

[38] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 11 030–11 039.

[39] Y. Wang, L. Deng, T. Zhang, and X. Wu, "Ssconv: Explicit spectral-to-spatial convolution for pansharpening," in *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2021, pp. 4472–4480.

[40] S. Peng, L. Deng, J. Hu, and Y. Zhuo, "Source-adaptive discriminative kernels based network for remote sensing pansharpening," *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2022.

[41] T. Zhang, L. Deng, T. Huang, J. Chanussot, and G. Vivone, "A triple-double convolutional neural network for panchromatic sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[42] Z. Jin, T. Zhang, T. Jiang, G. Vivone, and L. Deng, "Lagconv: Local-context adaptive convolution kernels with global harmonic bias for pansharpening," *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[43] D. Haase and M. Amthor, "Rethinking depthwise separable convolutions: how intra-kernel correlations lead to improved mobilenets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[44] L. Deng, G. Vivone, C. Jin, and J. Chanussot, "Detail injection-based deep convolutional neural networks for pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6995–7010, 2021.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017, pp. 5998 – 6008.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[47] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 568–578.

[48] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 1833–1844.

[49] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 184–199.

[50] X. Wu, T. Huang, L. Deng, and T. Zhang, "A decoder-free transformer-like architecture for high-efficiency single image deraining," in *IJCAI*, 2022.

[51] J. Xiao, X. Fu, A. Liu, F. Wu, and Z. Zha, "Image de-raining transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2022.

[52] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[53] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 576–584.

[54] X. Zhang, W. Huang, Q. Wang, and X. Li, "Ssr-net: Spatial–spectral reconstruction network for hyperspectral and multispectral image fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5953–5965, 2020.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[56] L. Xiangyu, L. Qingjie, and W. Yunhong, "Remote sensing image fusion based on two-stream fusion network," *Information Fusion*, vol. 55, pp. 1–15, 2020.

[57] Q. Xie, M. Zhou, Q. Zhao, Z. Xu, and D. Meng, "Mhf-net: An interpretable deep network for multispectral and hyperspectral image fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 1457–1473, March 2022.

[58] W. Dong, C. Zhou, F. Wu, J. Wu, G. Shi, and X. Li, "Model-guided deep hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, pp. 5754–5768, 2021.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[61] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 558–567.

[62] Z. Huang, Y. Ben, G. Luo, P. Cheng, G. Yu, and B. Fu, "Shuffle transformer: Rethinking spatial shuffle for vision transformer," *arXiv preprint arXiv:2106.03650*, 2021.

[63] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context for semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 8, pp. 2375–2398, 2021.

[64] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," in *International Conference on Learning Representations (ICLR)*, 2017.

[65] S. Li, R. Dian, L. Fang, and J. M. Bioucas Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4118–4130, 2018.

[66] Q. Wei, N. Dobigeon, and J. Y. Tourneret, "Fast fusion of multi-band images based on solving a sylvester equation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4109–4121, 2015.

[67] M. Selva, B. Aiazzi, F. Butera, L. Chiarantini, and S. Baronti, "Hyper-sharpening: A first approach on sim-ga data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3008–3024, 2015.

[68] R. Dian, S. Li, and X. Kang, "Regularizing hyperspectral and multispectral image fusion by cnn denoiser," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1124–1135, 2020.