A Model-Inspired Approach With Transformers for Hyperspectral Pansharpening

Yanli Shang, Jianjun Liu, Member, IEEE, Jinlong Yang, and Zebin Wu, Senior Member, IEEE

Abstract—Hyperspectral pansharpening is a method of integrating a low-resolution hyperspectral image with a high-resolution panchromatic image to produce a high-resolution hyperspectral image. In recent years, a number of hyperspectral pansharpening methods have been developed by using convolutional neural networks. However, these methods only consider local information due to limitations on the size of convolution kernels in the convolution operation. In this article, we design a model-inspired approach with transformers for hyperspectral pansharpening. Owing to the representation ability of transformers and the algorithmic explanatory ability of optimization models, our method is able to explore intrinsic relationships at a global scale on both the spectral and spatial features. First, we formulate an optimization model consisting of a fidelity term and a regularization term. Then, this optimization model is solved by a half-quadratic splitting algorithm and, thus, divided into two suboptimization problems: fidelity and regularization problems. Finally, the algorithm is implemented by a transformer-based deep structure. Specifically, the fidelity problem is solved by a gradient descent algorithm further and then implemented through a convolutional network. The regularization problem is depicted by an approximation operation and implemented via a transformer network. The experimental results on different satellite datasets demonstrate the effectiveness of the proposed method.

Index Terms—Half-quadratic splitting (HQS) algorithm, hyperspectral pansharpening, model driven, transformer.

I. INTRODUCTION

REMOTE sensing images are mainly acquired by various types of optical sensors carried by satellites or other space platforms. Owing to the different imaging strategies of the different sensors, panchromatic (PAN), multispectral (MS), and hyperspectral (HS) images can be obtained. PAN images provide excellent spatial resolution, but only one band of spectral information. HS and MS images are rich in spectral information but have poor spatial resolution. It is impractical to capture images with both high spectral and spatial resolutions owing to the limitations of imaging equipments. Therefore, image fusion techniques [1], [2], [3], [4] have been developed. For example, fusing low-resolution HS (LRHS) images with high-resolution PAN images can produce high-resolution HS (HRHS) images. This technique is called HS pansharpening, making the best use of spatial and spectral information to improve the efficiency of subsequent tasks like classification [5], [6], [7], detection [8], and tracking [9].

In the past few decades, researchers have proposed a variety of classical methods to enhance HS pansharpening techniques. They can be roughly divided into three categories: component substitution (CS) [10], [11], [12], [13], multiresolution analysis (MRA) [14], [15], and model-based [16], [17], [18], [19], [20], [21] and deep-learning-based [22], [23], [24], [25], [26], [27] methods. The goal of CS methods is to replace part or all of the spatial information of HS images with PAN images. For example, those exploiting intensity–hue–saturation [10], principal component analysis [11], Gram–Schmidt (GS) [12], and its enhanced version called GS adaptive (GSA) [13]. These methods are computationally fast, but the fusion results suffer from spectral distortion. The MRA methods are to inject the spatial detail information from PAN images into interpolated HS images, including the algorithms such as smoothing–filter–based intensity modulation [14] and wavelet transforms [15].

The model-based methods consider the fusion process as an ill-posed inverse optimization problem. It usually uses prior knowledge of remote sensing images to formulate hypotheses and construct spectral fidelity and structural fidelity terms, respectively. Examples are Laplace priors [16], variational priors [17], [18], [19], nonlocal priors [20], and low-rank priors [21]. Optimization solutions for model-based methods are usually based on iterative optimization algorithms. Examples include the gradient descent algorithm [28], [29], the split Bregman iteration algorithm [30], the half-quadratic splitting (HQS) algorithm [31], and the alternating direction method of multipliers (ADMM) algorithm [32]. The final results of the optimization algorithm are fused images. The model-based methods ensure that the spatial and spectral information is mostly retained in the fused image, which is more frequently applied to image fusion.

Deep-learning-based approaches have been applied to pansharpening, showing satisfactory performance due to their remarkable feature representing ability. It usually builds an end-to-end convolutional neural network (CNN) structure to learn the nonlinear relationship between inputs and outputs, by unsupervised [33] or supervised [22], [23], [24], [25], [26], [27] ways. Recently, the transformer-based methods have attracted the attention of researchers because the multthead self-attention

Manuscript received 18 April 2022; revised 21 June 2022 and 20 July 2022; accepted 12 August 2022. Date of publication 17 August 2022; date of current version 6 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62071204 and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20201338. (Corresponding author: Jianjun Liu.)

Yanli Shang, Jianjun Liu, and Jinlong Yang are with the Jiangsu Provincial Engineering Laboratory for Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214126, China (e-mail: aysy81@126.com; liuofficial@163.com; yjlgedeng@163.com).

Zebin Wu is with the School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zebin.wu@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2022.3199207

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
mechanism in the transformer differs from the structure of CNNs. The methods have been implemented for image classification [34], detection [35], segmentation [36], and super-resolution [37], achieving outstanding results.

There are still problems in the existing deep-learning-based methods for HS pansharpening. On the one hand, CNN-based approaches have been intensively studied, but there are shortcomings. The limitations of the convolutional kernel size and stride make it impossible for the CNN-based methods to fully extract the global features of an image. On the other hand, there are few studies based on a transformer, and its corresponding methods usually lack physical interpretation. Directly applying the original transformer to pansharpening may degrade its performance as redundant information in the images is easily taken into account during the calculation of self-attention. In addition, limited by the HS images, the transformer is faced with the challenge of the computational consumption of HS pansharpening.

To address the above problems, we propose a model-inspired approach with transformers for HS pansharpening (MTNet). First, we build an optimization model for the PAN and LRHS images, including a fidelity term and a regularization term. Second, we iteratively solve the optimization model using the HQS algorithm, where the fidelity term can be solved using a gradient descent algorithm and the regularization term can be replaced by a proximal operation. Finally, we design a transformer network to learn the iterative steps of the algorithm and obtain the final result. Specifically, the solutions of the fidelity terms are designed as a CNN structure, and the proximal operation is learned by a gradual sampling transformer network that can reduce computational costs. This design makes MTNet have the physical implications of HS pansharpening. In summary, the contributions of MTNet are described as follows.

1) We propose a novel HS pansharpening method that combines the model-driven deep technique and the transformer to ensure the interpretability of network design and robustness of network performance.

2) We add a linear mapping layer and a residual connection to the encoder module of the original transformer so that the gradient explosion and vanishing problems can be addressed, resulting in stable performance.

3) The MTNet adopts a gradual sampling strategy to address the nontrivial computational burden, which can not only capture global information but also reduce the computational complexity of the self-attention mechanism in the transformer.

The rest of this article is organized as follows. Section II briefly introduces the work related to HS pansharpening. Section III details the proposed optimization model, solution algorithm, and network structure. Section IV presents experiments on different data to validate the superiority of our proposed method. Finally, Section V concludes this article.

II. RELATED WORKS

In this section, we provide a brief overview of the HS pansharpening methods regarding model-based and deep-learning-based approaches. And the application of the transformer module in computer vision is also introduced.

A. Model-Based Approaches

Model-based methods can be roughly divided into nonfactorization-based and factorization-based methods. The nonfactorization-based approach acquires the target image from the entire perspective and using relevant a priori knowledge to achieve the desired result. For example, Ballester et al. [17] achieved the fusion of low-resolution MS and PAN images by a linear degradation of high-resolution MS (HRMS) images along the spatial and spectral dimensions. Based on this, Fu et al. [38] proposed a local gradient constraint to improve the performance of the fusion. Wei et al. [39] proposed a fast fusion method based on the Sylvester equation (FUSE), which decreased the computational complexity. The factorization-based methods focus on splitting the target image into two parts and regenerating the fused image through recovery techniques. It is roughly classified into matrix-based and tensor-based methods. For instance, Li and Yang [40] used a sparse representation, assuming that the signals of a remote sensing image are sparse in the basis set, applied to pansharpening. In addition, some improvement versions [41], [42] were suggested to produce better fusion results and to make the model more practical. Yokoya et al. [43] completed the fusion of LRHS and HRMS images using the matrix-factorization-based technique [44], [45], [46], considering that each spectral signature of an HRHS image can be mathematically represented as a linear combination of several endmembers, i.e., an HRHS image can be expressed as an endmember matrix multiplied by an abundance matrix. Lanaras et al. [47] obtained the two matrices by employing a projected gradient method. Because an HS image is essentially a 3-D tensor, the tensor-based methods are able to express hidden spatial structure and spectral information. For example, Dian et al. [48] represented an HS image as a tensor and obtained the HRHS image by a sparse core tensor and a dictionary of three modes. Since then, some improved versions have emerged [49], [50], [51]. Xu et al. [52] presented a nonlocal patch tensor sparse representation method to achieve the fusion of LRHS images with HRMS images. Besides, a new tensor factorization called tensor ring decomposition was suggested in [53]. In general, model-based methods have the ability to retain both spectral information and spatial information, but some model-based methods have complex and computationally intensive processes for solving optimization problems.

B. Deep-Learning-Based Approaches

Recently, deep-learning-based approaches, especially CNNs, have been developed to explore the nonlinear relationships between features and applied in the field of image processing with significant success. Typical instances are super-resolution [54], [55], [56], target detection [57], image segmentation [58], and image classification [59], [60]. Dong et al. [61] first proposed a super-resolution CNN (SRCNN) model, which is a milestone for the CNN in the field of image super-resolution. Inspired by the SRCNN, Masi et al. [23] proposed an SRCNN-based pansharpening that takes preinterpolated LRMS images and PAN images as inputs and learns the mapping relationship between input images and output HRMS images. Following the success of
CNNs in pansharpening, a variety of techniques have been invented to improve the performance of CNN-based methods, for instance, residual techniques [22], [26], [27], attention mechanisms [25], [62], detail injection [63], [64], etc. Although CNN-based methods have excellent feature extraction capabilities, they lack a physical interpretation of pansharpening.

Subsequently, many researchers came up with model-based deep learning methods to improve the interpretability of pure deep networks. These approaches are a combination of a traditional optimization model and a deep-learning-based approach. Specifically, an optimization model uses a task-specific prior as the regularization term, and an iterative algorithm is applied to implement the iterative process. Then, a deep network is designed for simulating the process of its solution. These methods are used in many areas, such as image deraining [65], image deblurring [66], and image super-resolution [67], [68], [69], [70], [71]. For example, Xie et al. [69] proposed a deep unfolding network to achieve the fusion of LRHS and HRMS images. Dong et al. [70] proposed a model-guided deep convolutional network that takes the observation matrix of HS images into account in the end-to-end optimization process, achieving excellent results. Zheng et al. [72] developed an edge-conditioned feature transform network for LRHS and HRMS, which uses an edge mapping prior to learn features of images adaptively. Shen et al. [68] presented a model for the fusion of LRHS and HRMS images solved by the ADMM algorithm, whose solution is unfolded using a deep network. Zheng et al. [71] used an unsupervised deep network to implement the matrix decomposition model. In the fields of pansharpening, the above methods cannot ensure the same outstanding results. To tackle this issue, Xu et al. [73] introduced a model for pansharpening with a deep approach, building two optimization problems whose solutions were unfolded by two network blocks, which were stacked alternately to form a deep network. However, these methods do not fully exploit the global information of HS, resulting in limited improvements.

C. Transformer Module

Before the emergence of transformers, most sequence models were based on CNNs or recurrent neural networks (RNNs). Two problems with sequential models are as follows: 1) sequential models can only be computed sequentially, which limits the ability to compute in parallel and there is information loss during the computation; and 2) the sequential model cannot solve the long-term dependence problem. The structure of a transformer consists of a self-attention module and a feedforward neural network. The structure breaks the limitations of structures of CNNs and RNNs, using self-attention mechanisms to learn the relationship between inputs and outputs from a global perspective and to achieve parallel computation. Based on the above, the transformer has been successfully applied to natural language processing (NLP) [74].

The transformer is growing at a remarkable rate and has become mainstream in the NLP field, gradually progressing into the field of computer vision [34], [35], [36], [75], [76], [77], [78], [79]. For example, a new model, Vision Transformer [34], was first applied to image classification with excellent results. Based on this, a number of methods were proposed to tackle the problem of image super-resolution. A learning texture transformer network for image super-resolution was proposed by Yang et al. [37]. The model efficiently retrieves and migrates high-definition texture information to maximize the use of reference image information. An appropriate migration of high-definition textures into the generated super-resolution results eliminates texture blurring and texture distortion. But this model cannot be designed too deep because of its heavy computational cost and high GPU memory occupation. To address this problem, Lu et al. [80] presented a novel efficient super-resolution transformer for fast and effective image super-resolution.

III. PROPOSED METHOD

A. Optimization Model and Algorithm

HS pansharpening is the process of fusing an LRHS image with a PAN image. For convenience, $Y \in \mathbb{R}^{B \times mn}$ represents an LRHS image with a height of $m$, a width of $n$, and a band number of $B$. $X \in \mathbb{R}^{B \times MN}$ represents a target HRHS image with a height of $M$, a width of $N$, and a band number of $b$. $P \in \mathbb{R}^{b \times MN}$ represents a PAN image, i.e., a single-band image ($b = 1$) that is equal in spatial resolution to $Y$ by a factor of $1/r$ ($r = M/m = N/n$). The relationship between the observation and target images can be written as:

$$Y = XB + NY$$  \hspace{1cm} (1)$$

$$P = RX + N_P$$  \hspace{1cm} (2)$$

where $B \in \mathbb{R}^{MN \times mn}$ represents the spatial blur and downsampling operation and $R \in \mathbb{R}^{b \times B}$ represents the spectral response function of the imaging sensor. $NY$ and $N_P$ denote the noises contained in $Y$ and $P$, respectively. To obtain $X$, we can solve the following optimization problem:

$$\min_X \frac{1}{2} \| Y - XB \|^2_F + \frac{\lambda}{2} \| P - RX \|^2_F.$$  \hspace{1cm} (3)$$

where $\| \cdot \|_F$ represents the Frobenius norm, and $\lambda > 0$ is used to balance the two fidelity terms. By referring to the work in [81], we retain the spectral fidelity term and replace the spatial fidelity term with a spatial regularization term denoted by $\phi$. Then, problem (3) can be rewritten as follows:

$$\min_X \frac{1}{2} \| Y - XB \|^2_F + \frac{\lambda}{2} \phi(X, P).$$  \hspace{1cm} (4)$$

To facilitate the use of the HQS\(^1\) algorithm to solve problem (4), a variable $H$ is introduced. Then, the optimization problem (4) can be rewritten as follows:

$$\min_X \frac{1}{2} \| Y - XB \|^2_F + \frac{\lambda}{2} \phi(H, P) \quad \text{s.t.} \quad H = X.$$  \hspace{1cm} (5)$$

\(^1\)For the optimization problem (4), it can be solved not only using the HQS algorithm but also the ADMM algorithm, the proximal gradient algorithm, etc. HQS is a simple and effective way to make the unfolding of a deep network more concise, so we have chosen it.
The constrained problem (5) is transformed into an unconstrained optimization problem as follows:

$$L_u(X, H) = \frac{1}{2} \| Y - XB \|_F^2 + \frac{\lambda}{2} \| X - H \|_F^2$$

(6)

where $\mu > 0$ is the penalty parameter. For problem (6), we update one variable at each step and fix the other, and repeat the update alternately. For $k = 1, 2, 3, \ldots, K$, we can repeat the following steps:

$$X^{(k)} = \arg\min_X L_u(X, H^{(k-1)})$$

$$= \arg\min_X \frac{1}{2} \| Y - XB \|_F^2 + \frac{\mu}{2} \| X - H^{(k-1)} \|_F^2$$

(7)

$$H^{(k)} = \arg\min_H L_u(X^{(k-1)}, H)$$

$$= \arg\min_H \frac{\mu}{2} \| X^{(k-1)} - H \|_F^2 + \frac{\lambda}{2} \varphi(H, P)$$

(8)

Problem (7) is convex, but we solve it by the gradient descent algorithm for the purpose of network implementation. It is described as follows:

$$X^{(k)} = X^{(k-1)} - \alpha(X^{(k-1)}B - Y)B^T - \alpha\mu(X^{(k-1)} - H^{(k-1)})$$

$$= X^{(k-1)} - t_1G^{(k-1)} - t_2(X^{(k-1)} - H^{(k-1)})$$

$$= (1 - t_2)X^{(k-1)} - t_1G^{(k-1)} + t_2H^{(k-1)}$$

(9)

where $t_1 = \alpha, t_2 = \alpha\mu$, and $G^{(k-1)} = (X^{(k-1)}B - Y)B^T$ are put in to facilitate the construction of network structure. Problem (8) can be solved when its regularization term is manually assigned. Recently, it has become popular to implement the regularization term via a deep prior [82], [83]. It is more adaptable and robust than traditional methods and facilitates the constructing of an end-to-end network structure. Therefore, problem (8) is solved by a proximal operation as follows:

$$H^{(k)} = \text{prox}_\varphi \left( X^{(k-1)}, \frac{\lambda}{2\mu} \right)$$

(10)

The entire process of solving the above problem (4) is detailed in Algorithm 1.

### Algorithm 1: HQS Algorithm for Problem (4).

1: **Input:** LRHS image $Y$, PAN image $P$, $B$, $R$

2: Initialize $H^{(0)}$ by matrix 0 and $X^{(0)}$ by bicubic interpolation of $Y$

3: for $k = 1 : K$ do

4: Update $X$ by (9)

5: Update $H$ by (10)

6: end for

7: **Output:** $X$

### Algorithm 2: Entire Procedure for MTNet.

1: **Input:** LRHS image $Y$, PAN image $P$

2: Initialize $X^{(0)}$ by bicubic interpolation of $Y$, $t_1$ by 0.1 and $t_2$ by 0.1

3: Initialize $G^{(0)}$ and $H^{(0)}$ by matrix 0

4: for $k = 1 : K$ do

5: $X^{(k)} = (1 - t_2)X^{(k-1)} - t_1G^{(k-1)} + t_2H^{(k-1)}$

6: $G^{(k)} = Encoder_{Decoder}(X^{(k-1)}, Y)$

7: $H^{(k)} = GSNet(X^{(k-1)}, P)$

8: end for

9: **Output:** $X^{(target)} = X^{(K)}$

### B. Network Structure

We propose MTNet to implement the fusion of HS and PAN images by unfolding the iterative process of Algorithm 1 as each layer of MTNet (see Fig. 1). MTNet consists of $K$ TransBlocks, corresponding to $K$ iterations of Algorithm 1. Each TransBlock [see Fig. 2(a)] includes the Encoder–Decoder module [see Fig. 2(b)], the gradual sampling network (GSNet) module [see Fig. 2(c)], which represent the two iteration steps of Algorithm 1. The inputs of each TransBlock are $Y$, $P$, and $X$, and the outputs are the updated variables $X$ and $H$, which are also the inputs to the next stage. The result of the last iteration of the MTNet is the desired HS image $X^{(target)}$, i.e., $X^{(K)} = X^{(target)}$. The entire process of MTNet is detailed in Algorithm 2.

1) **Encoder–Decoder:** In problem (9), we use a gradient descent algorithm to achieve an iterative update of $X$. 
This approach has similarities with Gaussian pyramid and Laplace structures, both of which require first simulating a Gaussian fuzzy downsampling and subsequently performing an inverse operation. The difference is that we cannot directly use spatial downsampling as \( B \), which does not guarantee the blindness of our proposed method. We may implement the local subtraction in problem (9) by using a neural network in a similar way to residual learning. Specifically, for \( G^{(k-1)} = (X^{(k-1)}B - Y)B^T \), we can think of \( (X^{(k-1)}B - Y) \) as the process of encoding and \( B^T \) as the process of decoding [see Fig. 2(b)]. \( B \) represents the spatial blur and downsampling operator, which is conducted using a 2-D strided convolution followed by BatchNorm and LeakyReLU. \( B^T \) can be considered as an inverse operation of \( B \), which is performed using a 2-D strided deconvolution. To accommodate the resolution, the stride is fixed to \( r \), and the kernel size is set to \( (r+1) \times (r+1) \). The Encoder–Decoder structure is not only consistent with the mathematical solution of problem (9), but also allows the local information to be extracted using the local perception capability of the convolutional network.

2) GSNet Module: GSNet is a gradual sampling network structure, which plays an important role in the overall network and is responsible for learning the proximal operation in problem (10). This module achieves the update of \( H \) in Algorithm 1. It mainly consists of two parts, the Down-transformer [see Fig. 3(a)] and the Up-transformer [see Fig. 3(b)]. The Down-transformer and the Up-transformer are formed by stacking transformers. Since the complexity of the transformer is quadratic to the spatial size, it is unwise to use the image directly as input to the transformer in HS pansharpening. It is also infeasible to cut the image into small patches as input to the transformer like [34], which would result in the loss of spatial information and poor quality of the fused image. Motivated by progressive sampling in image super-resolution [84], GSNet uses the transformer to build a progressive upsampling and downsampling network structure. The downsampling consisting of three Down-transformers is divided into three stages. The Down-transformer is applied to retrieve the shallow global information of each stage. The convolution is added later to capture the local information and edge information missed during the Down-transformer extraction. The upsampling consisting of two Up-transformers is divided into two stages. The Up-transformer is applied to extract the deeper global information. Moreover, skip connections and concatenation are applied in GSNet. This allows the dependencies between stages to be learned and enhances the ability of GSNet to acquire features. The GSNet module alleviates to a certain extent the high computational effort of the transformer and ensures the quality of the images generated by the whole network.

As can be seen in Fig. 3, the Down-transformer and the Up-transformer have the same transformer module [see Fig. 3(c)].
Fig. 3. Illustration of the GSNet module. (a) Down-transformer. (b) Up-transformer. (c) Transformer module used in the Down-transformer and Up-transformer. \( \oplus \) stands for addition.

In contrast to the original transformer [74], where only the encoder part is used, we add a linear layer and a residual connection. The linear layer is used to map the dimensions to output the dimensions we desire. The residual connection is able to improve the performance of the transformer by eliminating the problem of information loss caused by the deeper layers. In addition, “\(N \times \)” is the depth of the transformer and “Norm” represents LayerNorm, which is widely used in transformer-based methods, ensuring stability in training. “Feed Forward” represents a fully connected feedforward network, comprising two linear mapping layers and a LeakyReLU activation function [see Fig. 3(c)].

Multihead attention plays an important role in the transformer. It is a self-attention mechanism with multiple heads that enables the network to capture and takes into account various relationships from a global perspective. The typical process of a self-attention is as follows:

\[
Q = W_q X, K = W_k X, V = W_v X
\]

\[
\text{Attention} = \text{softmax} \left( \frac{X W_q W_k^T X^T}{\sqrt{d_k}} \right) X W_v
\]

\[
= \text{softmax} \left( \frac{Q K^T}{\sqrt{d_k}} \right) V \tag{11}
\]

where \(X\) denotes the input; \(Q, K,\) and \(V\) are the query, key, and value matrices, respectively; \(W_q, W_k,\) and \(W_v\) are the parameter matrices; and \(d_k\) is the query/key dimension. And the process of a multihead attention is as follows:

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, head_2, \ldots, head_h) W^O
\]

\[
head_i = \text{Attention}(Q W^Q_i, K W^K_i, V W^V_i) \tag{12}
\]

where \(h\) indicates the number of heads, and \(W^Q_i \in \mathbb{R}^{d_{\text{model}} \times d_k}, W^K_i \in \mathbb{R}^{d_{\text{model}} \times d_k},\) \(W^V_i \in \mathbb{R}^{d_{\text{model}} \times d_k},\) and \(W^O \in \mathbb{R}^{h d_k \times d_{\text{model}}}\) indicate the parameter matrices. \(d_{\text{model}}\) is the dimension of \(Q, K,\) and \(V \) of a single attention function, which is projected to \(d_k\) and \(d_v\) dimensions by the multihead self-attention. In (13), the inputs are divided into \(h\) groups of specific dimensions, and the results are concatenated by performing attention calculations on each group. The final multihead attention value is obtained by performing a linear transformation \(W^O.\) A more visual depiction of what has been stated above is shown in Fig. 4.

Notably, both the Down-transformer and the Up-transformer have a flatten operation before the data are fed into the transformer network (see Fig. 3). The flatten operation unfolds the data cube into a matrix so that each vector of the matrix represents the spectral information of the data cube. Then, the correlation of spatial information can be learned through the transformer. In the Down-transformer, we use average pooling as a downsampling operation to reduce the number of parameters and computations. In the Up-transformer, we use deconvolution as the upsampling operation.

In order to measure the difference between the network output \(X(\text{target})\) and the reference image \(X,\) we choose the \(l_1\) norm as the loss function. This is because it is more stable to outliers than the \(l_2\) norm. Then, the final objective function can be written as

\[
\text{Loss} = \| X - X(\text{target}) \|_1 \tag{13}
\]
where $\| \cdot \|_1$ is a commonly used regression loss function that calculates the mean of the absolute values of the differences between the targets and predicted values.

IV. EXPERIMENTAL RESULTS

In this section, we conducted experiments on three synthetic datasets and one real dataset to validate the effectiveness of MTNet.

A. Datasets

1) CAVE: The CAVE dataset consists of 32 indoor HS images of $512 \times 512$ pixels acquired in the laboratory environment, which contain 31 spectral bands acquired at wavelength intervals of 10 nm in the range of 0.4–0.7 μm. The training set of the CAVE dataset contains the first 20 images selected, and the remaining 12 images are used for testing.

2) Pavia Center (PC): The PC dataset is an image of the city acquired by the Reflection Optical System Imaging Spectrometer (ROSIS). The ROSIS sensors have a spectral range of 0.43–0.86 μm. Thirteen noise bands were removed, leaving 102 spectral channels. The HS image had a size of 1096 × 1096 pixels, but the central area contained no information and had to be discarded. Therefore, two subimages of 1096 × 223 and 1096 × 492 pixels, respectively, were produced. In the experiment, we used only the top left corner of the image, measuring 960 × 640 × 102. Then, the selected part was divided into 24 nonoverlapping cubes of size 160 × 160 × 102, which constituted the PC dataset. A sample of 16 cubes was randomly selected for training, and the remaining 12 images are used for testing.

3) Botswana: The Botswana dataset is a scene of Botswana Okavango Delta, South Africa, collected by the Hyperion sensor on NASA EO-1 satellite, with a spatial resolution of 30 m. This image initially had 1496 × 256 pixels with 242 bands covering spectral range of 400–2500 nm with a resolution of 10 nm. After removing the uncalibrated and noisy bands, the remaining 145 bands were used for the study. Our experiments utilized only the upper left corner portion of an image with a spatial coverage of 1200 × 240, which was then divided into 20 cubic patches of size 120 × 120 × 145, with no overlap. Fourteen cube patches were randomly chosen for training and the rest for testing.

4) University of Houston (UH): The UH dataset is a real dataset released by the 2018 Data Fusion Contest of the IEEE Geoscience and Remote Sensing Society, which contains an HRMS image (RGB image) of size 83 440 × 24 040 × 3 and an LRHS image of size 4172 × 1202 × 48. The PAN image is acquired by band averaging of HRMS images. For simplicity and consistency, we resize the PAN image to size 33 376 × 9616 × 1. Thus, the ratio between the PAN image and the LRHS image is 8. We crop three nonoverlapping 1024 × 1024 subimages from the PAN image and correspondingly three nonoverlapping 128 × 128 subimages from the LRHS image. We use the first two for training and the last one for testing.

According to Wald’s protocol, we generated the LRHS and PAN images by using the given HS images as reference. The LRHS images were acquired by Gaussian blurring and downsampling the HS reference images, where the Gaussian filter with a mean of 0 and a standard deviation of 2 was used, and the sampling factor was 8 ($r = 8$). The PAN images were generated by averaging the spectral bands in the visible range of the HS reference images. For the UH dataset, which has no reference image, we used the original LRHS image as the training label and downsampled the LRHS and PAN images for training. where $B$ and $R$ were estimated using the method in [67]. For testing, we used the original LRHS and PAN images as input to predict the final fused image.

B. Quality Measures

To assess the quality of the fusion results, eight popular metrics are used in this article. The first five are used for the synthetic dataset and the last three for the real dataset. For convenience, $X$ represents the reference image, $T$ represents the target image, $P$ represents the PAN image, and $Y$ represents the LRHS image, where $X(i,j,:)$ is the $(i,j)$th pixel of $X$ and $X(:,; t)$ is the $t$th band of $X$.

1) Spectral Angle Mapper (SAM) [85]: The SAM indicates the spectral quality of the fused image. The smaller the number of degrees, the better the spectral quality. The optimal value is 0.

$$\text{SAM}(X, T) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \arccos \left( \frac{X(i,j) \cdot T(i,j)}{\|X(i,j)\|_2 \|T(i,j)\|_2} \right)$$

(14)

2) Peak Signal-to-Noise Ratio (PSNR) [43]: The PSNR was used to measure the average spatial similarity between the target image and the reference image for all the bands. The higher the value, the lower the spatial distortion. The optimal value is $\infty$.

$$\text{PSNR}(X, T) = \frac{1}{B} \sum_{t=1}^{B} \text{PSNR}(X(t), T(t))$$

(15)

3) Structural Similarity Index Measure (SSIM) [3]: SSIM calculates the average structural similarity in the spatial domain between the generated image and the reference image. The higher the value of SSIM, the more similar the spatial structure. The preferred value is 1.

$$\text{SSIM}(X, T) = \frac{1}{B} \sum_{t=1}^{B} \text{SSIM}(X(t), T(t))$$

(16)

4) Root Mean Squared Error (RMSE) [3]: RMSE is used to represent the difference between the target image and the reference image. The smaller the difference, the better the result. The ideal value is 0.
RMSE(X, T)  
= \sqrt{\frac{1}{BMN} \sum_{i=1}^{B} \sum_{j=1}^{M} \sum_{t=1}^{N} (X(i, j, t) - T(i, j, t))^2}. \quad (17)

5) Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS) [3]: ERGAS is a global metric that reflects the overall quality of the fusion. The optimal value of ERGAS is 0, and the lower the value, the better the overall quality.

\[ \text{ERGAS}(X, T) = \frac{100}{r} \sqrt{\frac{1}{B} \sum_{t=1}^{B} \frac{\text{MSE}(T(t), X(t))}{\text{MEAN}^2(T(t))}} \quad (18) \]

where MSE is the mean square error function and MEAN is the mean value function. \( r \) is the sampling factor.

6) \( D_s[2] \): \( D_s \) is a spatial distortion index calculated as:

\[ D_s = \sqrt{\frac{1}{B} \sum_{t=1}^{B} \left| Q(T(t), P) - Q(Y(t), \tilde{P}) \right|^q} \quad (19) \]

where \( q \) is usually set to 1 and \( \tilde{P} \) is the spatial degradation of \( P \) whose spatial resolution is the same as \( Y \). The \( Q \) index [86] is calculated as the dissimilarity between \( Y \) and \( T \).

7) \( D_s[2] \): \( D_s \) is a spectral distortion index calculated as:

\[ D_s = \sqrt{\frac{1}{B(B-1)} \sum_{t_1=1}^{B} \sum_{t_2=1, t_2 \neq t_1}^{B} \left| Q(Y(t_1), Y(t_2)) - Q(T(t_1), T(t_2)) \right|^p} \quad (20) \]

where \( p \) is usually set to 1 and \( Q^{t_1, t_2} = Q(Y(t_1), Y(t_2)), Q^{t_1, t_2} = Q(T(t_1), T(t_2)).

8) Quality With No Reference (QNR) [2]: QNR is the product of the one’s complements of the spatial and spectral distortion indices, each raised to a real-valued exponent that contributes the relevance of spatial and spectral distortions to the overall quality. The two exponents jointly determine the nonlinearity of response in the interval [0, 1], same as a gamma correction, to achieve a better discrimination of the fusion results compared. It is calculated as

\[ \text{QNR} = (1 - D_s)\alpha(1 - D_s)\beta \quad (21) \]

where \( \alpha \) and \( \beta \) are two coefficients.

C. Compared Methods and Implementation Details

Seven methods were compared with our proposed methods, including the classical methods GSA [13],\(^6\) HySure [46],\(^7\) FUSE [39],\(^8\) CNMF [43],\(^9\) and the deep-learning-based methods PanNet [22],\(^9\) DARN [25],\(^10\) Fusion-Net [64],\(^11\) DBDNet [62],\(^12\) and GPPNN [73].\(^13\) The compared methods followed their papers and the corresponding source codes.

In preparing the training samples, overlapping patches of 48 × 48 pixels were extracted from the reference image as training samples. The strides of the extracted patches were 16 for the CAVE dataset, 4 for the PC dataset, and 3 for the Botswana dataset. The spatial sizes of the LRHS image and PAN image were 6 × 6 and 48 × 48 pixels, respectively. Approximately 20% of these extracted patches were used for validation. Our method was implemented using the PyTorch framework. It was optimized by AdamW for 300 epochs with a learning rate of 0.002 and a batch size of 16. The other parameters were given the default values in the AdamW algorithm.

In problem (9), there are two parameters \( t_1 \) and \( t_2 \) that are used to balance the results of each iteration. These two parameters are set as learnable parameters in the TransBlock, initialized \( t_1 = t_2 = 0.1, \) to obtain robust and satisfactory fusion results.

D. Parameter Selection and Ablation Studies

In this section, we will evaluate the number of different iterations \( K \) for all the datasets using a set of experiments. We have also demonstrated the effectiveness of our proposed method through ablation studies, in which the default parameters and structure are maintained.

1) Selection of \( K \): \( K \) is the number of iterations in Algorithm 1, which governs the depth of our proposed network. A proper value of \( K \) has a positive effect on the results of the MTNet. We conducted comparative experiments on the three datasets regarding the number of different network iterations and the corresponding spatial and spectral effects. The average quantitative results for the proposed method are presented in Table I with the best results marked in bold. Obviously, as the number of iterations increases, the performance of MTNet is enhanced on the three datasets. The SAM metric is optimal on the CAVE dataset when \( K = 3 \), and the remaining metrics are optimal at \( K = 5 \). This indicates that the effect is steady when \( K = 5 \). There are small differences in the results for all the indicators across different iterations of the PC dataset. Results are best when \( K = 2 \) for all the indicators. The ERGAS metrics are highest on the Botswana dataset when \( K = 4 \). The SSIM results are optimal when \( K = 3, 4, 5 \), and the RMSE results are optimal at \( K = 3, 5 \). Apart from ERGAS, the results for the other metrics are best when \( K = 5 \). In summary, as the number of iterations increases, the overall performance gets better both spatially and spectrally, but the trend slows down. Therefore, we set \( K = 5, 2, 5 \) for the CAVE, PC, and Botswana datasets, respectively.

\(^6\)The code is derived from the toolbox on the website: https://openremotesensing.net/wp-content/uploads/2017/11/HSMSFusionToolbox.zip.

\(^7\)Available: https://github.com/alfaiate/HySure.

\(^8\)Available: https://naotoyokoya.com/assets/zip/CNMFMATLAB.zip.

\(^9\)Available: https://github.com/oyam/PanNet-Landsat.

\(^10\)Available: https://github.com/yxzheng24/IEEE_TGRS_DHPDARN.

\(^11\)Available: https://github.com/shuangxd96/GPPNN.
2) Ablation Studies: As can be observed from Fig. 1, our proposed network consists of several TransBlock modules. TransBlock is replaced by ResNet, which has become more popular in recent years, to measure its importance. Specifically, we make ablation experiments to evaluate the effect of both the modules. "ResNet" means that the TransBlock module is replaced with a ResNet module in the MTNet. Figs. 5 and 6 show the performance gap between the two modules for different numbers of iterations. We can notice that whichever module is applied, the PSNR and SAM results improve as the number of iterations increases. This demonstrates the effectiveness of the proposed framework when the TransBlock module is replaced; the entire network can still extract features at each stage and inject feature information into the fused image. Furthermore, we can see that the results of "ResNet" on PSNR and SAM increase with the number of iterations, but the improvement is very small compared to “MTNet,” and there is even a decrease.

Overall, the difference between “ResNet” and “MTNet” is large, demonstrating the superiority of TransBlock in feature extraction and its importance in the structure of MTNet.

MTNet employs a gradual sampling strategy to reduce computational complexity, and an ablation experiment is proposed to demonstrate the effectiveness of this strategy. For convenience, “MTNet-G” represents the use of a gradual sampling strategy. “MTNet-N” represents the lack of using this strategy. Fig. 7 and Table II present the results on the CA VE dataset. Specifically, PARAMS and FLOPs are the results for an LRHS image of size $6 \times 6 \times 3$ and a PAN image of size $48 \times 48 \times 1$. TIME is the inference time for testing an LRHS image of size $8 \times 8 \times 31$ and a PAN image of size $64 \times 64 \times 1$. SAM, PSNR, SSIM, RMSE, ERGAS, and TIME in Table II are the results when the number of iterations $K = 1$. The results of FLOPs in Fig. 7 and Table II show that MTNet achieves satisfactory performance at
TABLE III
QUANTITATIVE METRICS OF THE COMPARED METHODS ON THE CAVE DATASETS

| Method | SAM | PSNR | SSIM | RMSE | ERGAS |
|--------|-----|------|------|------|-------|
| Best Values | 0 | +∞ | 1 | 0 | 0 |
| GSA | 11.0946 | 32.1207 | 0.8936 | 0.0257 | 3.2022 |
| HySure | 10.3809 | 34.6956 | 0.9161 | 0.0199 | 2.3624 |
| FUSE | 8.0570 | 30.6718 | 0.9035 | 0.0302 | 4.0564 |
| CNMF | 8.8773 | 31.2395 | 0.9205 | 0.0297 | 3.5475 |
| PanNet | 10.8846 | 34.1778 | 0.9420 | 0.0208 | 2.4882 |
| DARN | 4.7066 | 40.0616 | 0.9785 | 0.0109 | 1.3242 |
| Fusion-Net | 5.2753 | 39.1126 | 0.9725 | 0.0124 | 1.4703 |
| DBDENet | 6.1151 | 36.5469 | 0.9578 | 0.0161 | 1.8787 |
| GPPNN | 4.9003 | 38.5315 | 0.9762 | 0.0134 | 1.6042 |
| MTNet | 3.8854 | 41.1732 | 0.9812 | 0.0100 | 1.1808 |

E. Results of Experiments

In order to comprehensively evaluate the performance of our proposed methods, seven methods mentioned in Section IV-C are used for comparison. The results of the three datasets are described qualitatively and visually in the following.

1) CAVE: Table III shows the average results of 12 test images on the five quality evaluation metrics, with the best results marked in bold. Apparently, our proposed method is superior to the compared methods in all the metrics. This demonstrates that the MTNet method can accomplish HS pansharpening with the least distortion. In addition, the proposed MTNet method outperforms the five CNN-based HS pansharpening approaches, which further validates the superiority of transformer modules.

The competition between the reconstructed images and the corresponding error images on the “real_and_fake_peppers” of the CAVE dataset is shown in Fig. 8. The reconstructed image is from the 20th band of the fused image, and the error image represents the difference between the reconstructed image and the ground truth. A meaningful region of each reconstructed image is marked and zoomed in four times. From the reconstructed images, the GSA, HySure, FUSE, and CNMF methods have different degrees of loss of detail information, while the remaining methods are relatively close to the ground truth [see Fig. 8(k)]. The error images for the conventional methods, PanNet and DBDENet, differ significantly from the ground truth, while those for DARN, Fusion-Net, GPPNN, and MTNet are closer, but more similar for MTNet. In addition to this, a comparison of the fusion quality in different spectral bands is shown in Figs. 9(a) and 11(a). Overall, the spectral curves of the traditional methods differ significantly from the reference curves. The remaining methods differ less from it. The PSNR curves of these methods are presented, and the PSNR values of our proposed MTNet are higher than those of other methods in all the bands. This further proves the effectiveness of MTNet on the CAVE dataset.

2) PC: Table IV lists the average results of the eight test images of 160 × 160 size on the PC dataset. In terms of overall results, the differences between the compared methods are not very significant due to the relatively small training samples in the PC dataset. In the SSIM, DBDENet is the best, and Fusion-Net and MTNet are outperforming the rest of the methods. In the SAM, MTNet is the best, and Fusion-Net and GPPNN are superior to the remaining methods. PanNet, DARN, Fusion-Net, GPPNN, and our MTNet show better results in PSNR, RMSE, and ERGAS metrics compared to the remaining methods. However, our MTNet method performs the best.

Fig. 10 shows the comparison of the reconstructed images and the corresponding error images for all the methods. By looking at the reconstructed images, we can conclude that those of GSA, HySure, FUSE, and DARN have varying degrees of brightness errors. This proves that these methods have poor performance in terms of spectral fidelity. The error images for PanNet, GPPNN, and MTNet are relatively close. However, the reconstructed images of PanNet and GPPNN methods have contour blurring, i.e., there is a problem of loss of edge information. Fig. 9(b) shows the spectral curves for all the methods on the PC dataset. It can be seen that the MTNet curve overlaps the most with the reference. Fig. 11(b) shows the PSNR versus spectral band for all the methods on the PC dataset. Owing to the complex band information of the PC dataset, there is also a higher content of topographic information reacting between the bands. In 1 and 80–85 bands, most methods have difficulty in fusing well. However, the PanNet, DARN, DPPNN, and MTNet methods still maintain excellent results. This demonstrates the robustness of these methods and their ability to recover spectra. In general terms, our method achieves higher PSNR values than other methods, showing that our method retains the spectral information well on the PC dataset.

3) Botswana: For the Botswana dataset, Table V lists the average results of all the competing HS pansharpening methods. We can see that all the methods show relatively excellent results, but our method performs best. Fig. 12 shows the comparison between reconstructed images and error images on all the methods. It can be seen that the reconstructed images...
from CNMF and PanNet are blurry. This indicates that the CNMF and PanNet methods lack the ability to preserve spectral information when applied to the Botswana dataset. Detailed information is lost in the reconstructed images of GSA and DARN, while those of HySure, FUSE, GPPNN, and MTNet are closer to ground truth [see Fig. 12(k)]. The error images of GSA, HySure, FUSE, CNMF, and PanNet differ significantly from the ground truth, while DARN, Fusion-Net, DBDENet, DPPNN, and MTNet are closer, especially MTNet. Fig. 9(c) shows the spectral curves for all the methods on the Botswana dataset. DARN, Fusion-Net, DBDENet, GPPNN, and MTNet are closer to the reference, while the remaining methods differ slightly.
Fig. 10. Visual results of PC dataset at band 44. Top row: reconstructed images (with a meaningful region marked and zoomed in two times for easy observation). Bottom row: error images. (a) GSA. (b) HySure. (c) FUSE. (d) CNMF. (e) PanNet. (f) DARN. (g) Fusion-Net. (h) DBDENet. (i) GPPNN. (j) MTNet. (k) Ground truth.

Fig. 11. PSNR as a function of spectral band. (a) CAVE dataset. (b) PC dataset. (c) Botswana dataset.

from it. The PSNR curves for Botswana are given in Fig. 11(c). All the methods perform well on all the bands, but our method has the best PSNR results. In a summary, the experimental results and visual analysis show that the proposed MTNet method can achieve excellent pansharpening performance on the Botswana dataset.

4) UH: For the UH dataset, Table VI lists the average results of all the competing HS pansharpening methods. It can be seen that all the methods show relatively good results. FUSE is the best on $D_5$ metrics, and MTNet is the best on $D_6$ and QNR. Fig. 13 illustrates the fusion of UH visually, and it can be seen that HySure, GPPNN, and MTNet all give better results. However, MTNet is the best on the contours of the image. In general, the results of MNet on the UH dataset demonstrate its superiority in terms of quantitative metrics and visual appearance, illustrating the good generalization capability of MTNet.
Fig. 12. Visual results of the Botswana dataset at band 52. Top row: reconstructed images (with a meaningful region marked and zoomed in two times for easy observation). Bottom row: error images. (a) GSA. (b) HySure. (c) FUSE. (d) CNMF. (e) PanNet. (f) DARN. (g) Fusion-Net. (h) DBDENet. (i) GPPNN. (j) MTNet. (k) Ground truth.

Fig. 13. Visual results of the UH dataset. (a) Real LRHS image. (b) Real PAN image. (c) GSA. (d) HySure. (e) FUSE. (f) CNMF. (g) PanNet. (h) DARN. (i) Fusion-Net. (j) DBDENet. (k) GPPNN. (l) MTNet.

F. Computational Efficiency

To clarify the computational efficiency of our proposed method, we discuss three evaluation metrics of each competing method on the CAVE datasets. For the training phase, all the competing methods are implemented with PyTorch and run on a single GeForce GTX 3060 12-GB graphic card. Specifically, Table VII shows the results of the experiment on the CAVE dataset. “PARAS” and “FLOPs” are the results for an LRHS image of size $6 \times 6 \times 31$ and a PAN image of size $48 \times 48 \times 1$. “TIME” is the inference time for testing a PAN image of size $64 \times 64 \times 1$ and an LRHS image of size $8 \times 8 \times 31$. Obviously, MTNet has the smallest trainable parameters, DARN performs best in FLOP computations, and Fusion-Net takes the fewest inference times. In general, MTNet is relatively expensive in
V. CONCLUSION

This article proposes a model-inspired deep approach with transformers for HS pansharpening. The approach mainly combines the traditional model-based method with the transformer-based method. For the model-based method, there are a data fidelity term and a regularization term. It is solved iteratively by an HQS algorithm and is decomposed into two suboptimization problems, namely, the data fidelity problem and the regularization problem. For the data fidelity problem, it is solved by a gradient descent algorithm. For the regularization problem, it is expressed by a proximal operation. Using the unfolding technique, the two optimization problems are implemented by two network modules. The Encoder–Decoder module performs the data fidelity problem by means of convolutional operations, and the GSNet module learns the proximal operation using the transformer network. Specifically, the transformer is an evolution of the encoder part of the original transformer by adding a linear mapping layer and a residual connection. This allows it to be more effective in feature extraction and information supplementation. GSNet builds a gradual sampling structure network using transformers. This structure enhances the performance of fusion and solves the problem of the computationally intensive self-attention in transformers. The Encoder–Decoder and GSNet modules are assembled into a TransBlock module. The stacking of several TransBlocks forms the MTNet corresponding to the iterative steps of the algorithm. The final fusion result is the output of the last TransBlock of MTNet. Experimental results on three simulated datasets demonstrated that our method outperformed state-of-the-art methods in terms of quality and quantity.

In future work, we will consider further enhancements on the transformer to reduce GPU consumption even more. In addition, the existing network could be enhanced more effectively to further improve performance.

ACKNOWLEDGMENT

The authors would like to thank the authors of [13], [22], [25], [39], [43], [46], [62], [64], and [73] for providing their codes, Prof. P. Gamba from the University of Pavia for providing the PC dataset, and the anonymous reviewers for their constructive comments on this article. The authors would also like to thank the National Centre for Airborne Laser Mapping and the Hyperspectral Image Analysis Lab.

REFERENCES

[1] H. Ghassemian, “A review of remote sensing image fusion methods,” Inf. Fusion, vol. 32, pp. 75–89, 2016.
[2] G. Vivone et al., “A critical comparison among pansharpening algorithms,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 5, pp. 2565–2586, May 2015.
[3] L. Loncan et al., “Hyperspectral pansharpening: A review,” IEEE Geosci. Remote Sens. Mag., vol. 3, no. 3, pp. 27–46, Sep. 2015.
[4] R. Dian, S. Li, B. Sun, and A. Guo, “Recent advances and new guidelines on hyperspectral and multispectral image fusion,” Inf. Fusion, vol. 69, pp. 40–51, 2021.
[5] J. Liu, Z. Wu, J. Li, A. Plaza, and Y. Yuan, “Probabilistic-kernel collaborative representation for spatial–spectral hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 4, pp. 2371–2384, Apr. 2016.
[6] J. Liu, Z. Wu, L. Xiao, and H. Yan, “Learning multiple parameters for kernel collaborative representation classification,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 12, pp. 5068–5078, Dec. 2020.
[7] J. Liu, Z. Wu, J. Li, X. Sun, and H. Yan, “Generalized tensor regression for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 2, pp. 1244–1258, Feb. 2020.
[8] D. W. Stein, S. G. Beaver, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, “Anomaly detection from hyperspectral imagery,” *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, Jan. 2002.

[9] H. Van Nguyen, A. Banerjee, and R. Chellappa, “Tracking via object recognition using a hyperspectral video camera,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 44–51.

[10] T.-M. Tu, P. Huang, C.-L. Hung, and C.-P. Chang, “A fast intensity-hue-saturation fusion technique with spectral adjustment for IKONOS imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 309–312, Oct. 2004.

[11] M. G. Liao, N. H. Younan, and R. L. King, “An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1323–1335, May 2008.

[12] 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.

[13] B. Aiazzi, S. Baronti, and M. Selva, “Improving component substitution pansharpening through multivariate regression of MS + pan data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3230–3239, Oct. 2007.

[14] J. 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, 2000.

[15] J. Zhou, D. L. Civeo, and J. Silander, “A wavelet transform method to merge landslide TM and SPOT pansharpenic data,” *Int. J. Remote Sens.*, vol. 19, no. 4, pp. 743–757, 1998.

[16] H. Shen, X. Meng, and L. Zhang, “An integrated framework for the spatio-temporal–spectral fusion of remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7153–7178, Dec. 2016.

[17] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé, “A variational model for PaXS image fusion,” *Int. J. Comput. Vis.*, vol. 69, no. 1, pp. 43–58, 2006.

[18] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, “A new pansharpening algorithm based on total variation,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 318–322, Jan. 2014.

[19] R. Molina, M. Vega, J. Mateos, and A. K. Katagergilas, “Variational posterior distribution approximation in Bayesian super resolution reconstruction of multispectral images,” *Appl. Comput. Harmon. Anal.*, vol. 24, no. 2, pp. 251–267, 2008.

[20] J. Duran, A. Buades, B. Coll, and C. Sbert, “A nonlocal variational model for pansharpening image fusion,” *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 761–796, 2014.

[21] S. Yang, K. Zhang, and M. Wang, “Learning low-rank decomposition for pan-sharpening with spatial-spectral offsets,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3647–3657, Aug. 2018.

[22] 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.*, 2017, pp. 5449–5457.

[23] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, “Pansharpening by convolutional neural networks,” *Remote Sens.*, vol. 8, no. 7, 2016, Art. no. 594.

[24] R. Dian, S. Li, A. Guo, and L. Fang, “Deep hyperspectral image sharpening,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5345–5355, Nov. 2018.

[25] Y. Zheng, J. Li, Y. Li, J. Guo, X. Wu, and J. Chanussot, “Hyperspectral pansharpening using deep prior and dual attention residual network,” *IEEE Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8059–8076, Nov. 2020.

[26] L. Liu et al., “Shallow–deep convolutional network and spectral-discrimination-based detail injection for multispectral image pan-sharpening,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, no. 1, pp. 1783–1790, Feb. 2020.

[27] W. Wei, Z. Lin, Y. Huang, and X. Ding, “A variational pan-sharpening method with local gradient constraints,” in *Proc. IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 5791–5800, 2019.

[28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[29] 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.

[30] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image representation at scale,” in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[31] K. Cheng, J. Du, H. Zhou, D. Zhao, and H. Qin, “Image super-resolution based on half quadratic splitting,” *Infrared Phys. Technol.*, vol. 105, 2020, Art. no. 103193.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[33] 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.

[34] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image representation at scale,” in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[35] K. Cheng, J. Du, H. Zhou, D. Zhao, and H. Qin, “Image super-resolution based on half quadratic splitting,” *Infrared Phys. Technol.*, vol. 105, 2020, Art. no. 103193.
[56] J. Yao, D. Hong, J. Chanussot, D. Meng, X. Zhu, and Z. Xu, “Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution,” in Proc. Eur. Conf. Comput. Vis., vol. 2020, pp. 208–224.

[57] D. Tan et al., “A circular target feature detection framework based on DCCNN for industrial applications,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 5, pp. 3303–3313, May 2021.

[58] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[59] H. Guo, J. Liu, Z. Xiao, and L. Xiao, “Deep CNN-based hyperspectral image classification using discriminative multiple spatial-spectral feature fusion,” Remote Sens. Lett., vol. 11, no. 9, pp. 827–836, 2020.

[60] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[61] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2016.

[62] J. Qu, S. Hou, W. Dong, S. Xiao, Q. Du, and Y. Li, “A dual-branch detail extraction network for hyperspectral pansharpening,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5518413.

[63] 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.

[64] 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.

[65] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in Proc. IEEE/CVF Conf. Vis. Pattern Recognit., 2020, pp. 3103–3112.

[66] Y. Li, M. Tofghi, J. Geng, V. Monga, and Y. C. Eldar, “Efficient and interpretable deep blind image deblurring via algorithm unrolling,” IEEE Trans. Comput. Imag., vol. 6, pp. 666–681, 2020.

[67] J. Liu, Z. Wu, L. Xiao, and X.-J. Wu, “Model inspired autoencoder for unsupervised hyperspectral image super-resolution,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5522412.

[68] D. Shen, J. Liu, Z. Wu, J. Yang, and L. Xiao, “ADMM-HFNet: A matrix decomposition-based deep approach for hyperspectral image fusion,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5513417.

[69] Q. Xie, M. Zhou, Q. Zhao, Z. Xu, and D. Meng, “MHI-Net: An interpretable deep network for multispectral and hyperspectral image fusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 3, pp. 1457–1473, Mar. 2022.

[70] W. Dong, C. Zhou, F. Wu, J. Wu, G. Shi, and X. Li, “Model-guided deep hyperspectral image super-resolution,” IEEE Trans. Image Process., vol. 30, pp. 5754–5768, 2021.

[71] K. Zheng et al., “Coupled convolutional neural network with adaptive response function learning for unsupervised hyperspectral super resolution,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 3, pp. 2487–2502, Mar. 2021.

[72] Y. Zheng et al., “Edge-conditioned feature transform network for hyperspectral and multispectral image fusion,” IEEE Trans. Geosci. Remote Sens., vol. 50, 2022, Art. no. 5513315.

[73] S. Xu, J. Zhang, Z. Zhao, K. Sun, J. Liu, and C. Zhang, “Deep gradient projection networks for pansharpening,” in Proc. IEEE/CVF Conf. Vis. Pattern Recognit., 2021, pp. 1366–1375.

[74] A. Vaswani et al., “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010.

[75] N. Parmar et al., “Image transformer,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4055–4064.

[76] Z. Sun, S. Cao, Y. Yang, and K. M. Kitani, “Rethinking transformer-based set prediction for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 3611–3620.

[77] Z. Dai, B. Cai, Y. Lin, and J. Chen, “UP-DETR: Unsupervised pre-training for object detection with transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1601–1610.

[78] A. Botach, E. Zelhemovszki, and C. Bajd, “End-to-end referring video object segmentation with multimodal transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 4985–4995.

[79] Y. Jiang, S. Chang, and Z. Wang, “TransGAN: Two pure transformers can make one strong GAN, and that can scale up,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, vol. 34, pp. 14745–14756.

[80] Z. Lu, J. Li, H. Lin, C. Huang, L. Zhang, and T. Zeng, “Transformerfor single image super-resolution,” in Proc. IEEE/CVF Conf. Vis. Pattern Recognit., 2022, pp. 457–466.

[81] R. C. Hardie, M. T. Eisemann, and G. L. Wilson, “MAP estimation for hyperspectral image resolution enhancement using an auxiliary sensor,” IEEE Trans. Image Process., vol. 13, no. 9, pp. 1174–1184, Sep. 2004.

[82] Y. Sun, J. Liu, J. Yang, Z. Xiao, and Z. Wu, “A deep image prior-based interpretable network for hyperspectral image fusion,” Remote Sens. Lett., vol. 12, no. 12, pp. 1250–1259, 2021.

[83] W. Xie, J. Lei, Y. Cui, Y. Li, and Q. Du, “Hyperspectral pansharpening with deep priors,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 5, pp. 1529–1543, May 2019.

[84] Y. Wang, F. Peruzzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, “A fully progressive approach to single-image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2018, pp. 864–873.

[85] 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. Summaries 3rd Annu. JPL Airborne Geosci. Work., 1992, pp. 147–149.

[86] Z. Wang and A. C. Bovik, “A universal image quality index,” IEEE Signal Process. Lett., vol. 9, no. 3, pp. 81–84, Mar. 2002.

Yanli Shang received the B.S. degree in network engineering from the Anyang Institute of Technology, Anyang, China, in 2020. She is currently working toward the master’s degree in electronic information with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. Her research interests include deep learning and hyperspectral image fusion.

Jianjun Liu (Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. From 2018 to 2020, he was a Postdoctoral Researcher with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong. His research interests include hyperspectral image classification, super-resolution, spectral unmixing, sparse representation, computer vision, and pattern recognition.

Jinlong Yang received the Ph.D. degree in pattern recognition and intelligence system from Xidian University, Xi’an, China, in 2012. He is currently an Associate Professor with Jiangnan University, Wuxi, China. His current research interests include target tracking, information fusion, and signal processing.

Zebin Wu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science, Nanjing University of Science and Technology. His research interests include hyperspectral image processing, high-performance computing, and computer simulation.