Hyperspectral Image Joint Super-Resolution via Local Implicit Spatial-Spectral Function Learning

Yanan Zhang, Jizhou Zhang, and Sijia Han

***Abstract*—Hyperspectral image (HSI) super-resolution (SR) in both spatial and spectral dimensions is one of the most attractive research topics in HSI processing. Although recent advances in deep learning (DL) frameworks have greatly improved the performance of spatial-spectral SR reconstruction, existing methods learn discrete representations of HSI, ignoring real-world signals' continuous nature. Recently, Implicit Neural Representation (INR) has been applied to 3D surface reconstruction and image SR for continuous representation and has attracted increasing attention. In this paper, we propose the Local Implicit Spatial-spectral Function (LISSF), which learns a local continuous representation of high spatial resolution hyperspectral images (HR-HSI) from the discrete inputs. The model consists of a deep feature encoder and a spatial-spectral intensity decoder. The encoder converts the low spatial resolution multispectral image (LR-MSI) into deep features and the decoder predicts the intensity values at the given coordinates as output. Since the spatial-spectral coordinates are continuous, LISSF can achieve spatial-spectral SR in arbitrary scales, even extrapolating to higher resolutions not covered by the training data. Extensive experiments on spatial-spectral SR, spatial SR, and spectral SR demonstrate that LISSF can achieve superior performance in comparison with state-of-the-art methods. Moreover, ablation studies are performed on the effects of individual components of LISSF.**

***Index Terms*—Hyperspectral image (HSI), spatial-spectral super-resolution, implicit neural representations (INR), local implicit spatial-spectral function (LISSF).**

I. Introduction

HPERSPECTRAL images (HSI) contain reflectance or transmittance information of objects in hundreds of spectral bands over a continuous wavelength range. Compared to commonly used RGB images, HSIs show more intrinsic properties of object materials. Therefore, hyperspectral imaging is an indispensable scientific tool in many fields such as remote sensing [1], [2], [3], medical imaging [4], [5], [6], and industrial inspection [7], [8], [9]. In these applications, both high spatial resolution and high spectral resolution are required. However, there is a trade-off between spatial and spectral resolution due to limitation of imaging sensor technology and time constraints. For example, multispectral imaging (MSI) systems on remote sensing satellites such as Geo Eye-1, MODIS, Landsat series, and GF series have lower spatial resolution than the panchromatic or RGB imaging systems, and lower spectral resolution than HSI systems. Neither the spectral resolution nor the spatial resolution of these MSI systems can satisfy the requirements of emerging remote sensing applications. In recent years, to take full advantage of the available multispectral data, deep learning (DL) frameworks have been introduced to enhance the resolution in spatial and spectral dimensions.

There are mainly three approaches for obtaining high spatial-spectral resolution data using DL-based SR methods: 1) spatial SR, 2) spectral SR, and 3) spatial-spectral SR. The spatial SR approach only enhances the spatial resolution of MSI. A typical paradigm is training a deep convolutional neural network (CNN) network to extract deep features from low spatial resolution inputs and reconstruct high spatial resolution counterparts. The spectral SR approach only enhances the spectral resolution of MSIs. Sparse representations and dictionary learning are commonly used conventional techniques for spectral SR, and CNN-based models have become popular in recent years. A special case of spectral SR is to reconstruct HSIs from RGB images, which has become an important task in many computer vision challenges such as NTIRE [10], [11]. Unlike the above approaches, the spatial-spectral SR approach extends both spatial resolution and spectral resolution of the input. Therefore, it can make better use of the available multispectral data and adapt to more situations. At the same time, it is a more challenging task due to its highly ill-posed nature.

Although several DL-based spatial-spectral SR methods have been proposed, these methods still suffer from a number of issues that hinder their performance. On the one hand, most existing methods of these approaches treats the hyperspectral image as discrete voxels in the 3D spatial-spectral space, ignoring the continuous nature of signals. On the other hand, existing methods are trained and inferred at a fixed SR scale which is inconvenient in practical use.

To address these issues and inspired by the recent progress in INR, we propose a framework, termed Local Implicit Spatial-spectral Function (LISSF), that learns the local continuous spatial-spectral representation of HR-HSI from discrete input. LISSF consists of two parts, an encoder and a decoder. The encoder is a transformer-based U-shape network to extract deep features from LR-MSI input. The decoder takes an MLP as the

Manuscript received 1 January 2024; revised 17 April 2024; accepted 30 April 2024. Date of publication 7 May 2024; date of current version 15 May 2024. This work was supported by China Postdoctoral Science Foundation under Grant BX20200051. *(Corresponding author: Yanan Zhang.)*

Yanan Zhang is with the School of Arts and Media, Hubei Business College, Wuhan 430079, China (e-mail: zhangyanan@hbc.edu.cn).

Jizhou Zhang is with the Mech-Mind Robotics Technologies Ltd., Beijing 100085, China (e-mail: xiaomianzhou@gmail.com).

Sijia Han is with the Laboratory of Microwave Sensing, National Space Science Center, Chinese Academy of Science, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: hansijia18@mails.ucas.ac.cn).

Digital Object Identifier 10.1109/JPHOT.2024.3397232

core and uses the deep features to estimate the intensity values at given continuous spatial-spectral coordinates on HR-HSI. As a spatial-spectral SR model, LISSF can improve the resolution of LR-MSI in both spatial and spectral dimensions. Furthermore, it can scale the LR-MSI input to arbitrary size regardless of whether the current scaling ratio is covered by the training process, which greatly improves the practicality.

To evaluate the performance of LISSF, detailed experiments on CAVE and ARAD HS datasets are carried out. Joint spatial-spectral SR experiments validate the state-of-the-art performance of LISSF. The well performance of spatial SR and spectral SR also demonstrates the good generalization ability and flexibility of LISSF. Moreover, ablation studies are carried out to validate the effectiveness of individual components of LISSF.

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

1) A novel framework termed LISSF is proposed for jointly spatial-spectral SR. To the best of our knowledge, this is the first work to learn local continuous representation of HSI for spatial-spectral SR.
2) For the first time, DL-based arbitrary resolution scaling in both spatial and spectral dimensions are achieved, which brings significant convenience to the practical application of hyperspectral image super-resolution.
3) Extensive experiments on spatial SR, spectral SR and spatial-spectral SR are conducted to compare the proposed model with state-of-the-art methods. A modified model based on MetaSR [12] which is capable of arbitrary spatial-spectral SR is used for comparison.

II. Releated Work

A. Spatial Super-Resolution

HSIs are data in 3D form, containing two spatial dimensions and one spectral dimension. Spatial SR, i.e. increasing the resolution of HSIs in the spatial dimensions, has the same roots as the single-image super-resolution (SISR) task in computer vision. Numerous SISR methods can be directly used to perform HSI spatial SR, such as SRCNN [13], VDSR [14], SRGAN [15] and EDSR [16]. However, these methods treat images of different bands as independent, ignoring their correlation. Motivated by these approaches, networks dedicated to HSI spatial SR have recently been developed. For the first time, Yuan et al. [17] proposed a CNN-based method for HSI spatial SR. They regarded the problem as a transfer learning task and transferred a pre-trained SISR model to perform the HSI spatial SR. Later, Mei et al. [18] proposed a 3D-FCNN model to directly increase the spatial resolution of HSI. Li et al. [19] designed a generative adversarial network (GAN) framework for HSI spatial SR to reconstruct more texture details and proposed a band attention mechanism to explore the correlation of spectral bands. Li et al. [20] designed a novel mixed with both 2D and 3D convolution to jointly exploit the information from different bands. Jiang et al. [21] proposed to use spatial-spectral blocks (SSB) to exploit the spatial and spectral prior. Liu et al. [22] employed a new spectral attention mechanism for group convolutions to rescale grouped features with holistic spectral information. Li et al. [23] alternately employed 2D and 3D units to solve the problem of structural redundancy by sharing spatial information during the reconstruction.

B. Spectral Super-Resolution

The spectral SR denotes to enhance the resolution of hyperspectral images in the spectral dimension. Most previous researches of spectral SR are based on the sparse representation. Han et al. [24] proposed a spectral library-based dictionary learning method to achieve HSI spectral SR, which estimates the band matching matrix, spectral dictionary, and sparse coefficients simultaneously. Yi et al. [25] designed a framework involving spectral improvement strategies and spatial preservation strategies for HSI spectral SR. In recent years, many CNN-based methods have been proposed and achieved excellent spectral SR performance. Gewali et al. [26] proposed to reconstruct HSIs from MSIs using an end-to-end fully convolutional residual neural network architecture. Arun et al. [27] integrated sparse representation into a CNN-based encoder-decoder architecture to improve the fidelity of spectral SR reconstruction. Zheng et al. [28] proposed a spatial-spectral residual attention network (SSRAN) that simultaneously explores the spatial and spectral information of MSIs to reconstruct HSIs. In particular, reconstruction of HSIs from RGB images can be regarded as a special case of spectral SR. It has become a hot topic in the field of computer vision [11], attracting the attention of many researchers. Shi et al. [29] proposed two advanced CNNs for RGB spectral SR, one using residual blocks and the other using dense blocks with a novel fusion scheme. Li et al. [30] proposed an adaptive weighted attention network (AWAN) for RGB spectral SR which integrats adaptive weighted channel attention (AWCA) module and patch-level second-order non-local (PSNL) module. Cai et al. [31] designed a Transformer-based method, Multi-stage Spectral-wise Transformer (MST++), for efficient spectral reconstruction. The model achieves state-of-the-art performance while consuming much less computation and memory.

C. Spatial-Spectral Super-Resolution

Although spatial SR and spectral SR have been widely explored in recent years, few researches consider joint spatial-spectral SR. For the first time, Mei et al. [32] proposed a spatial-spectral joint SR (SSJSR) model that learns an end-to-end mapping from a LR-MSI and to the HR-HSI with a full a 3-D CNN. Ma et al. [33] proposed a CNN-based model named unfolding spatiospectral super-resolution network (US3RN). US3RN solves both spatial SR and spectral SR problems via the alternative direction multiplier method (ADMM) technique. Ma et al. [34] presented a deep spatial-spectral feature interaction network (SSFIN) that using a spatial-spectral feature interaction block (SSFIB) to make the spatial SR task and the spectral SR task benefit each other. Our work also belongs to spatial-spectral SR methods. Moreover, it can achieve arbitrary SR in both spatial and spectral domains.

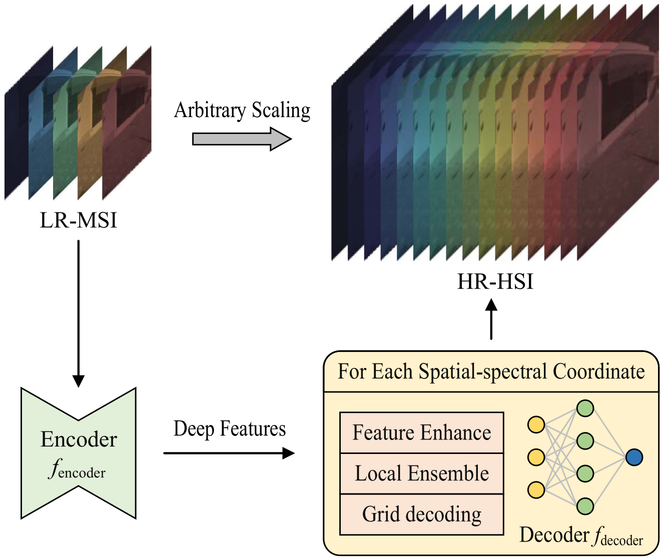

Fig. 1. Schematic diagram of Local Implicit Spatial-spectral Function (LISSF). LISSF learns the local continuous representation from discrete input and achieves arbitrary super-resolution in both spatial and spectral dimensions.

D. Implicit Neural Representation

INR is an emerging technology that can generate continuous, memory-efficient implicit representations for objects like shapes [35], [36], scenes [37], [38], [39] or images [40]. Objects are represented as multi-layer perceptrons (MLPs) that map coordinates to signal values. Genova et al. [35] chose an implicit surface representation based on a combination of local shape elements to allow for widely varying geometry and topology of shapes. Peng et al. [39] combined a convolutional encoder with an implicit occupancy decoder to incorporate inductive biases for structured reasoning in 3D space. Recently, many studies have focused on sharing the function space of implicit representations for different objects, rather than learning an independent INR for each object [41]. Sitzmann et al. [41] proposed a meta-learning-based method for sharing the function space. Chen et al. [42] proposed the Local Implicit Image Function (LIIF) to generate continuous representations for images and achieve arbitrary spatial scaling of images. Xu et al. [43] and Zhang et al. [44] proposed to represent hyperspectral images with INR and perform spectral reconstruction from RGB images. However, both studies focus on learning independent INRs and achieving SR in only spectral domain. Our work proposes to learn the local image function in a shared spatial-spectral space and can achieve arbitrary scaling in both spatial and spectral domains.

III. Methodology

To simplify the presentation, we use the abbreviations listed in Table I. In this section, we introduce the proposed LISSF model as shown in Fig. 2. The model takes a LR-MSI I as input and produce a HR-HSI O as output. The model is mainly divided into two parts: encoder and decoder. The encoder converts the input I into a 3D deep feature F_d through a deep neural network. Then, the decoder maps each continuous spatial-spectral coordinate $s \in S$ to the hyperspectral pixel value $O(s)$ using a MLP. Finally, all hyperspectral pixel values are synthesized and reshaped to generate O.

TABLE I
Abbreviations and Notations

Abbreviations	Description
LR-MSI	low spatial resolution multispectral image
LR-HSI	low spatial resolution hyperspectral image
HR-MSI	high spatial resolution multispectral image
HR-HSI	high spatial resolution hyperspectral image
I	LR-MSI $\in \mathbb{R}^{h\times w\times d}$
O	HR-HSI $\in \mathbb{R}^{H\times W\times D}$
F_d	3D deep feature $\in \mathbb{R}^{h\times w\times d\times C}$
S	Continuous spatial-spectral coordinates $\in \mathbb{R}^{HWD\times 3}$
s	Continuous spatial-spectral coordinate $\in \mathbb{R}^{1\times 3}$
s^*	Discrete spatial-spectral coordinate $\in \mathbb{R}^{1\times 3}$
t	Local feature vector $\in \mathbb{R}^{1\times C}$
$\hat{t}$	Enhanced feature vector $\in \mathbb{R}^{1\times 27C}$

A. Spatial-Spectral Feature Representation

Let $I \in \mathbb{R}^{h\times w\times d}$ denotes the input LR-MSI, where h, w and d represent the height, width and spectral bands respectively. The encoder transforms I into a 3D deep feature $F_d \in \mathbb{R}^{h\times w\times d\times C}$, where C represents the channel number. It can be formulated as

$$F_d = f_{\text{encoder}}(I), \tag{1}$$

where f_{encoder} is the map function of encoder.

In LISSF, we use a transformer-based network as the encoder, as shown in Fig. 2. To cope with different number of input spectral bands and to exploit the feature representation ability of 3D data, we use 3D convolution layers as basic components of the encoder. It first applies a convolution with kernel size of 3 to extract shallow features from the input, denoted as

$$F_s = f_{\text{conv3}}(I), \tag{2}$$

where f_{conv3} is the map function of 3D convolution with kernel size of 3. Afterwards, these shallow features are transformed into deep features through a 3-level U-shaped structure. In each level, multiple transformer blocks are stacked to effectively extract features. Starting from the original input, the encoder hierarchically reduces spatial size, while keeping spectral number and expanding channel size. The detailed process can be expressed as

$$F_1 = f_{\text{trans}}^{N_1}\left(\cdots f_{\text{trans}}^{1}\left(F_s\right)\right), \tag{3}$$

$$F_2 = f_{\text{trans}}^{N_2}\left(\cdots f_{\text{trans}}^{1}\left(f_{\text{down}}\left(F_1\right)\right)\right), \tag{4}$$

$$F_3 = f_{\text{trans}}^{N_3}\left(\cdots f_{\text{trans}}^{1}\left(f_{\text{down}}\left(F_2\right)\right)\right), \tag{5}$$

where f_{trans} is the map function of transformer block, f_{down} denotes the map function of downsampler, N_1, N_2 and N_3 are the number of transformer blocks in each level. $F_1 \in \mathbb{R}^{h\times w\times d\times C}$, $F_2 \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times d\times 2C}$ and $F_3 \in \mathbb{R}^{\frac{h}{4}\times\frac{w}{4}\times d\times 4C}$ are the deep features in different levels. Then, the encoder hierarchically expands spatial size, while keeping spectral number and reducing channel size, formulated as

$$F_4 = f_{\text{trans}}^{N_2}\left(\cdots f_{\text{trans}}^{1}\left(f_{\text{conv1}}\left(f_{\text{cat}}\left(F_2, f_{\text{up}}\left(F_3\right)\right)\right)\right)\right), \tag{6}$$

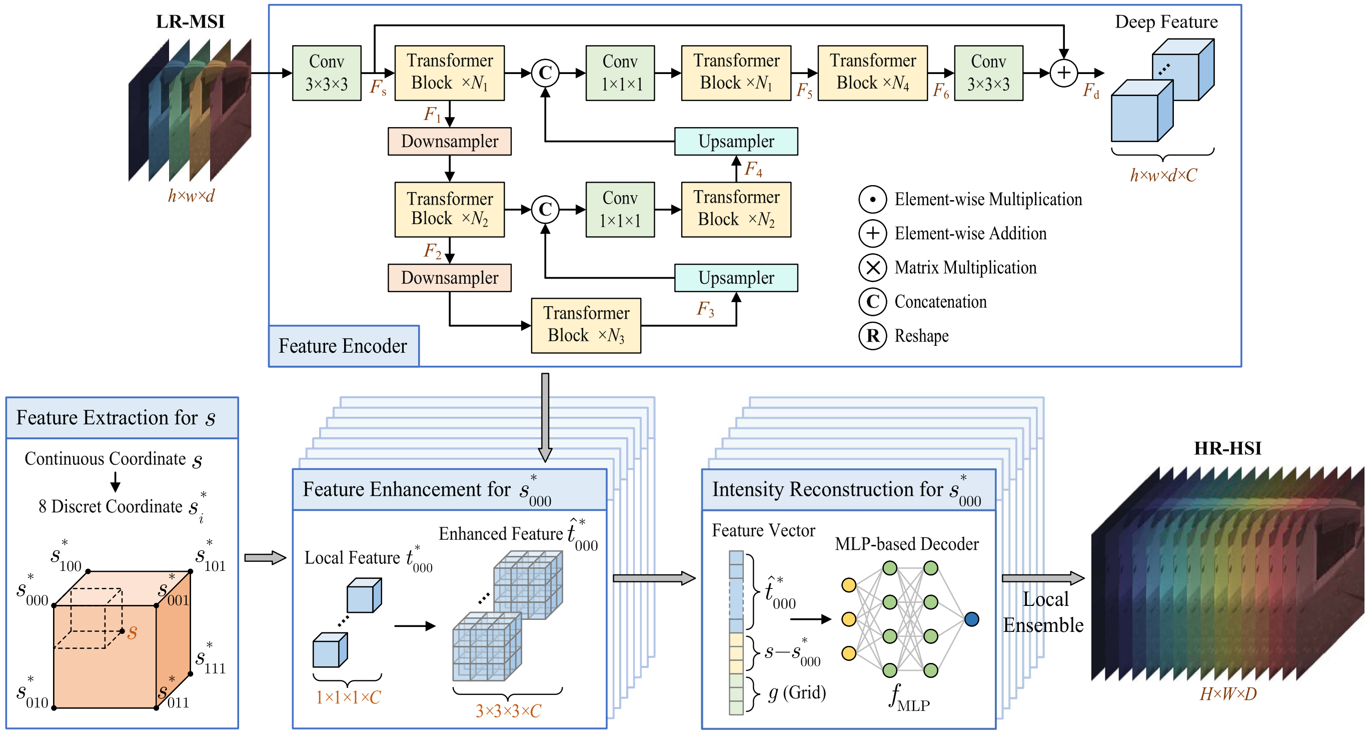

Fig. 2. Diagram of LISSF for spatial-spectral SR. First, the encoder, a transformer-based U-shape network, transforms the LR-MSI input into deep features. Then, for a specific continuous spatial-spectral coordinate, the decoder extracts a local feature and enhances it. At last,an MLP is used to estimate the corresponding intensity value. Besides, the local ensemble strategy is used to ensure a smooth reconstruction. .

$$F_5 = f_{\mathrm{trans}}^{N_1}\left(\cdots f_{\mathrm{trans}}^{1}\left(f_{\mathrm{conv1}}\left(f_{\mathrm{cat}}\left(F_1, f_{\mathrm{up}}\left(F_4\right)\right)\right)\right)\right), \quad (7)$$

where f_{up} denotes the map function of upsampler, f_{conv1} is the map function of 3D convolution with kernel size of 1, $F_4 \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times d\times 2C}$ and $F_5 \in \mathbb{R}^{h\times w\times d\times C}$ are the deep features after processing. Afterwards, multiple transformer blocks are stacked to refine the features, denoted as

$$F_6 = f_{\mathrm{trans}}^{N_4}\left(\cdots f_{\mathrm{trans}}^{1}\left(F_5\right)\right). \quad (8)$$

At last, a skip connection from the shallow features is used to generate the final deep features, formulated as

$$F_{\mathrm{d}} = F_{\mathrm{s}} + f_{\mathrm{conv3}}\left(F_6\right). \quad (9)$$

1) Transformer block: In typical transformer-based models, the transformer block is composed of a Multi-head Self-Attention (MSA) block, a Feed-Forward Network (FFN) block and the corresponding layer normalization blocks. To improve the performance of transformer block and inspired by the structures proposed in Restormer [45], we propose the 3D multi-dconv head transposed attention (MDTA3D) and the 3D gated-dconv feed-forward network (GDFN3D). The architecture of the transformer block is illustrated in Fig. 3. Suppose X as the input, the map function of the transformer block can be denoted as

$$f_{\mathrm{trans}}(X) = \hat{X} + f_{\mathrm{GDFN3D}}\left(f_{\mathrm{LN}}\left(\hat{X}\right)\right), \quad (10)$$

$$\hat{X} = X + f_{\mathrm{MDTA3D}}\left(f_{\mathrm{LN}}(X)\right), \quad (11)$$

where f_{MDTA3D} and f_{GDFN3D} are the map functions of MDTA3D and GDFN3D respectively.

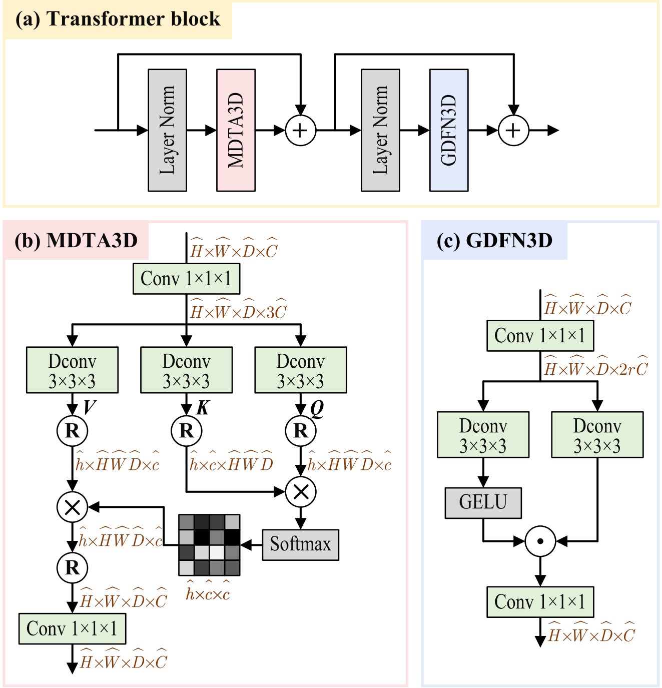
(a) Transformer block
Layer Norm
MDTA3D
Layer Norm
GDFN3D
(b) MDTA3D
$\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}$
Conv 1×1×1
$\hat{H}\times\hat{W}\times\hat{D}\times 3\hat{C}$
Dconv 3×3×3
Dconv 3×3×3
Dconv 3×3×3
V
K
Q
R
$\hat{h}\times\hat{H}\hat{W}\hat{D}\times\hat{c}$
$\hat{h}\times\hat{c}\times\hat{H}\hat{W}\hat{D}$
$\hat{h}\times\hat{H}\hat{W}\hat{D}\times\hat{c}$
Softmax
$\hat{h}\times\hat{H}\hat{W}\hat{D}\times\hat{c}$
$\hat{h}\times\hat{c}\times\hat{c}$
$\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}$
Conv 1×1×1
$\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}$
(c) GDFN3D
$\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}$
Conv 1×1×1
$\hat{H}\times\hat{W}\times\hat{D}\times 2r\hat{C}$
Dconv 3×3×3
Dconv 3×3×3
GELU
Conv 1×1×1
$\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}$

Fig. 3. Structure of (a) the transformer block, (b) the MDTA3D block, and (c) the GDFN3D block.

2) Channel-Wise Multi-Head Transposed Attention: Suppose $X \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}}$ as the input of MDTA3D, X is first projected into *query* ($Q \in \mathbb{R}^{\hat{h}\times\hat{H}\hat{W}\hat{D}\times\hat{c}}$), *key* ($K \in \mathbb{R}^{\hat{h}\times\hat{c}\times\hat{H}\hat{W}\hat{D}}$) and *value* ($V \in \mathbb{R}^{\hat{h}\times\hat{H}\hat{W}\hat{D}\times\hat{c}}$), where $\hat{h}$ is the head number and $\hat{c}$ is the channel number in each head. It is achieved by applying 1×1 convolutions to expand the channel dimension, and then using 3×3 group channel-wise convolutions to encode spatial-spectral context, formulated as

$$Q = f^{Q}_{\text{reshape}}\left(f_{\text{dconv3}}\left(f^{1}_{\text{chunk}}\left(f_{\text{conv1}}(X)\right)\right)\right), \tag{12}$$

$$K = f^{K}_{\text{reshape}}\left(f_{\text{dconv3}}\left(f^{2}_{\text{chunk}}\left(f_{\text{conv1}}(X)\right)\right)\right), \tag{13}$$

$$V = f^{V}_{\text{reshape}}\left(f_{\text{dconv3}}\left(f^{3}_{\text{chunk}}\left(f_{\text{conv1}}(X)\right)\right)\right), \tag{14}$$

where $f^{1}_{\text{chunk}}, f^{2}_{\text{chunk}}, f^{3}_{\text{chunk}}$ denote to split the feature into three chunks, f_{dconv3} is the map function of 3D deep-wise convolution with kernel size of 3, $f^{Q}_{\text{reshape}}, f^{K}_{\text{reshape}}, f^{V}_{\text{reshape}}$ are reshape functions corresponding to Q, K, V. Then the map function of MDTA3D can be expressed as

$$f_{\text{MDTA3D}}(X) = f_{\text{conv1}}\left(V\cdot f_{\text{softmax}}\left(K\cdot Q\alpha\right)\right), \tag{15}$$

where f_{softmax} is the softmax activation function. By using channel-wise multi-head attention mechanism and channel-wise group convolution, the memory usage and computation can be greatly reduced.

3) Channel-Wise Gated Feed-Forward Network: Suppose $X\in\mathbb{R}^{\hat{H}\times\hat{W}\times\hat{D}\times\hat{C}}$ as the input of GDFN3D, it is expanded in channel dimension and spit into two parallel paths to achieve gating mechanism. Then the gated feature is reduced to the original size in channel dimension. The map function of GDFN3D can be formulated as

$$f_{\text{GDFN3D}}(X) = f_{\text{conv1}}\left(f_{\text{gating}}\left(f_{\text{conv1}}(X)\right)\right), \tag{16}$$

$$l f_{\text{gating}}\left(\hat{X}\right) = f_{\text{dconv3}}\left(f^{1}_{\text{chunk}}\left(\hat{X}\right)\right) \odot f_{\text{dconv3}}\left(f_{\text{GELU}}\left(f^{2}_{\text{chunk}}\left(\hat{X}\right)\right)\right), \tag{17}$$

where f_{gating} denotes the map function of gating mechanism, f_{GELU} is the GELU (gaussian error linear units) activation function.

B. Continuous Spatial-Spectral Reconstruction

Let $O\in\mathbb{R}^{H\times W\times D}$ denotes the output HR-HSI, where H, W and D represent the height, width and spectral bands respectively. It is clear that $H\geq h$, $W\geq w$ and $D\geq d$ for the spatial-spectral SR task. In typical spatial SR and spectral SR models, the upsampling module is crucial for mapping the deep features from low-resolution space to high-resolution space. However, the scaling ratio of the upsampling module is fixed. To achieve arbitrary scaling in the spatial and spectral domains, we map each spatial-spectral continuous coordinate $s\in S$ to its corresponding hyperspectral pixel value $O(s)$ with the deep feature F, formulated as

$$O(s) = f_{\text{decoder}}(F, s), \tag{18}$$

where f_{decoder} is the map function of the decoder. Inspired by former research on Local Implicit Image Function (LIIF) [42], the reconstruction process of LISSF can be separated as the following four parts.

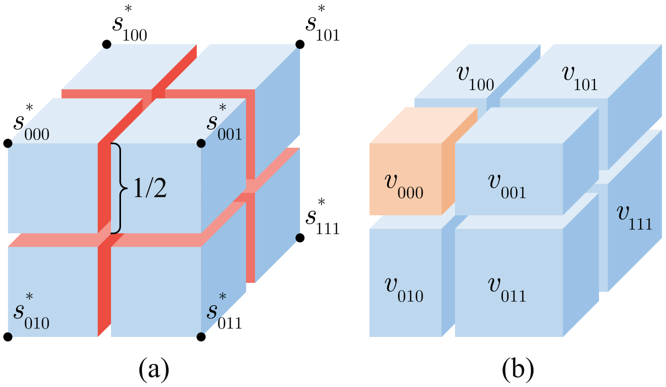

Fig. 4. LISSF with local ensemble. (a) 8 nearest discrete coordinates s^*_i of s and their interfaces. (c) Normalized volumes as local ensemble weights.

1) Local feature extraction: More specifically, to map the continuous coordinate $s\in S$ to HR-HSI value $O(s)$, the feature vector t needs to be extracted first. As the deep feature F is represented in the discrete space, t can be obtained by indexing F at the nearest (Euclidean distance) discrete coordinate s^*, formulated as

$$t = F\left(s^*\right). \tag{19}$$

Then (18) can be rewritten to

$$O(s) = f_{\text{decoder}}(F, s) = f_{\text{MLP}}\left(f_{\text{cat}}\left(t, s-s^*\right)\right), \tag{20}$$

where f_{MLP} is the mapping function of the MLP, f_{cat} refers to concatenation of vectors. In (20), t and $s-s^*$ make up the 1D input and can be transformed to $O(s)$ with the MLP. Using the residual value $s-s^*$ instead of s prevents the MLP from relying on the absolute value of s, allowing it to learn the local continuous representation.

2) Feature enhancement: Although (20) is enough to train the decoder, neighboring information is ignored with only one coordinate. To achieve a better representation of the local information, we apply the feature enhancement scheme. The deep features in the $3\times3\times3$ neighboring area are concatenated together to generate the final feature vector, formulated as

$$\hat{t} = f_{\text{cat}}\left(F\left(s^*+[l,m,n]\right)\right), \tag{21}$$

where $l\in\{-1,0,1\}$, $m\in\{-1,0,1\}$ and $n\in\{-1,0,1\}$ denote the variations of discrete spatial-spectral coordinates. After the feature enhancement, t is replaced by $\hat{t}$ in subsequent processes, so (20) can be rewritten to

$$O(s) = f_{\text{decoder}}(F, s) = f_{\text{MLP}}\left(f_{\text{cat}}\left(\hat{t}, s-s^*\right)\right). \tag{22}$$

3) Local ensemble: There is still an issue in (22) that hinders the continuous prediction of pixel values. Since the pixel value is predicted by querying the nearest feature vector with the decoder, when s moves across the boundary of adjacent discrete coordinates, the nearest discrete coordinate s^* and its corresponding feature vector $\hat{t}$ change, and the decoder's prediction changes accordingly. As illustrated in Fig. 4(a), the sudden switch happens when s crossing the red interfaces. As long as the decoder map function f_{decoder} is not perfect, discontinuous prediction can appear at these interfaces when the sudden switches occur. Therefore, we refine (22) to

$$\begin{aligned} O(s) &= f_{\text{decoder}}(F, s) \\ &= \sum_{i} \frac{v_i}{v} \cdot f_{\text{MLP}}\left(f_{\text{cat}}\left(\hat{t}_i, s - s_i^*\right)\right), \end{aligned} \tag{23}$$

where $s_i^*(i \in \{000, 001, 010, 011, 100, 101, 110, 111\})$ is one of the 8 nearest discrete coordinates of s, $\hat{t}_i$ is the corresponding enhanced feature vector of s_i^*, v_i is the volume of the subspace enclosed by s and s_i^*, $v = \sum_i v_i$ is the total volume of 8 subspaces. (23) denotes to combine 8 neighboring predictions of the decoder to get the final prediction. With the normalized volumes of 8 subspaces as weights, the final prediction can switch smoothly around the interfaces, as shown in Fig. 4(b).

4) Grid decoding: Since LISSF is used for arbitrary spatial-spectral SR, the decoder should exhibit different properties with different scaling ratio. For example, the model should tend to reconstruct more textures with lower ratio and reconstruct more low-frequency features with higher ratio. Therefore, taking scale-dependent information as additional input features will improve the reconstruction performance of the decoder. We finally extend (23) as

$$\begin{aligned} O(s) &= f_{\text{decoder}}(F, s) \\ &= \sum_{i} \frac{v_i}{v} \cdot f_{\text{MLP}}\left(f_{\text{cat}}\left(\hat{t}_i, s - s_i^*, g\right)\right), \end{aligned} \tag{24}$$

where $g = [g_h, g_w, g_d]$ specifies the grid size of the input data cube in spatial and spectral dimensions.

IV. Experiments and Analyses

A. Experimental Setups

1) Datasets: In the experiments, we use 2 hyperspectral datasets, CAVE [46] and ARAD HS (NTIRE 2020) [11]. CAVE is used in training and testing, while ARAD HS is used for testing only.

The CAVE [46] dataset consists of 32 hyperspectral images which covering varieties of materials and objects, such as skin, fruits, drinks, feathers, paintings, etc. The CAVE is captured by a tunable filter and a cooled CCD camera called Apogee Alta U260 under controlled indoor illumination conditions. All images are saved in 16-bit format to preserve high dynamic information. Each image has 31 spectral bands with a wavelength range of 400–700 nm with 10 nm interval and a spatial resolution of 512 × 512 pixels. 24 HSIs in the CAVE are used for traing and the other 8 are used for testing.

The ARAD HS dataset is built for the NTIRE 2020 challenge on spectral reconstruction from RGB images. It contains two parts: Track 1 "Clean" and Track2 "Real World", each of which contains 450 training images and 10 validation images. The ARAD HS dataset is collected with a Specim IQ mobile hyperspectral camera. Each image has 482 × 512 pixels and 31 bands from 400 nm to 700 nm with a 10 nm step. We use 10 validation image in the "Clean" Track for testing.

2) Implementation details: As the dataset contains only HR-HSI, we obtain the corresponding LR-MSI input by downsampling the HR-HSI. We use bicubic interpolation for spatial downsampling and linear interpolation for spectral downsampling. Due to the difference between spatial and spectral pixel definitions, we align corners when interpolating the spectral dimension, but not when interpolating the spatial dimensions. Therefore, the definition of magnification in the spectral dimension is different from that in the spatial dimensions. For example, ×2 represents 16 channels to 31 channels, and ×3 represents 11 channels to 31 channels. Since LISSF is cable of arbitrary SR in spatial and spectral dimensions, the scaling factors are not fixed during training. The spatial scaling factor is ranging from 1 to 4 and the spectral scaling factor is ranging from 1 to 5.

During the training phase, the input patch is of $48\times48\times6$ pixels. The ground truth patch is randomly cropped from the original HR-HSI and its size is determined by the scale factors. A total of 20 patches are randomly selected for training in one image. Image flip and rotation are randomly used for data augmentation. The proposed model and other methods for comparison are all trained for 200 epochs. We use the AdamW optimizer to train LISSF with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of 2×10^{-4}. The learning rate is initialed as 3×10^{-4} and halved every 20 epochs. For a fair comparison, the channel number of all methods are set as 64. The proposed model and other methods are all implemented using the PyTorch framework and trained on an NVIDIA GTX3090 GPU.

3) Evaluation metrics: To quantitatively evaluate the performance of the proposed method, we use three widely used metrics, including Peak Signal to Noise Ratio (PSNR), structural similarity (SSIM), and spectral angle mapping (SAM). PSNR is the ratio between the maximum possible power of an image and the power of distortion noise that affects the quality of its reconstruction. It is suitable for evaluating the overall reconstruction performance of different methods. SSIM is a perception-based model that considers image degradation as perceptual change in structural information. It is suitable to evaluate the spatial reconstruction performance of different methods. SAM determines the similarity between the estimated spectra and the reference one by calculating the angle between them. It is suitable to evaluate the spectral reconstruction performance of different methods.

4) State-of-the-art methods: To evaluate the performance of the proposed method under different conditions, we use three state-of-the-art spatial-spectral SR methods, SSJSR [32], US3RN [33] and SSFIN [34] for comparison. We also make up four spatial-spectral SR methods by combining state-of-the-art HSI spatial SR methods (MCNet [20] and ERCSR [23]) and HSI spectral SR methods (AWAN [30] and MST++[31]). The interpolation method that upsampling with bicubic interpolation in spatial dimensions and linear interpolation in spectral dimension is used as a baseline. Besides, we modify the MetaSR [12] method into a 3D form, making it possible to achieve spatial-spectral SR and compare with LISSF. The modified MetaSR3D model apply the same encoder as LISSF.

TABLE II
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF ALL METHODS ON CAVE DATASET

Methods	Spatial scale	Spectral scale	PSNR↑	SSIM↑	SAM↓
Interpolation	×2	×2 (16 to 31)	40.53	0.9863	6.194
AWAN [30] & MCNet [20]			41.50	0.9857	6.086
AWAN [30] & ERCSR [23]			42.79	0.9890	5.646
MST++ [31] & MCNet [20]			40.98	0.9839	6.305
MST++ [31] & ERCSR [23]			42.12	0.9870	5.913
US3RN [33]			40.28	0.9816	6.532
SSJSR [32]			33.73	0.9478	9.517
SSFIN [34]			40.23	0.9802	6.570
MetaSR 3D [12]			42.08	0.9588	7.445
LISSF (Ours)			**43.36**	**0.9897**	**5.466**
Interpolation	×4	×5 (7 to 31)	31.40	0.9245	10.878
AWAN [30] & MCNet [20]			34.94	0.9516	9.525
AWAN [30] & ERCSR [23]			**35.44**	0.9577	8.992
MST++ [31] & MCNet [20]			34.99	0.9524	9.481
MST++ [31] & ERCSR [23]			35.14	0.9527	9.444
US3RN [33]			35.16	0.9533	9.255
SSJSR [32]			29.90	0.8836	13.367
SSFIN [34]			34.55	0.9437	10.139
MetaSR 3D [12]			34.23	0.9477	9.818
LISSF (Ours)			35.02	**0.9637**	**8.632**

B. Spatial-Spectral SR Results on CAVE Dataset

1) Setup: To evaluate the spatial-spectral SR performance of LISSF, experiments are carried out on the CAVE dataset under two scaling factor settings. The first is to increase the spatial resolution by 2 times and the spectral resolution by 2 times. The second is to increase the spatial resolution by 4 times and the spectral resolution by 5 times. It should be noted that the LISSF and MetaSR3D model are trained only once while other models are retrained for different scales.

2) Results: For quantitative comparison, the average PSNR, SSIM and SAM metrics of all methods on the CAVE dataset are shown in Table II where bold indicates the best results and underline indicates the second best results. The interpolation method can be use as a baseline and perform well under low scaling ratios. Compared with the four combined methods, the three previously proposed state-of-the-art methods, SSJSR [32], US3RN [33] and SSFIN [34] have no obvious advantages. This may be because the four combined models have significantly larger network size than the three independent models. As can be seen from Table II, AWAN [30] & ERCSR [23] method achieves the best performance and LISSF achieves the second best of all methods. As mentioned earlier, LISSF is trained only once, while the rest of the algorithms are trained individually for each scale. For qualitative comparison, Fig. 5 provides examples of visual reconstruction for all methods under two scaling factor settings. The corresponding metrics are listed below the pictures and regions of interest (ROI) are magnified. It can be easily figured out that the results of LISSF contain abundant details and little reconstruction error. We also plot the spectral intensity of interested points in Fig. 6 where the reconstructed spectrum of LISSF is very close to the ground truth. All these experiment results demonstrate the effectiveness of the proposed LISSF method.

TABLE III
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF METASR 3D AND LISSF ON CAVE DATASET WITH ARBITRARY SCALING FACTORS

Methods	Spatial scale	Spectral scale	PSNR↑	SSIM↑	SAM↓
Interpolation	×2.5	×2.5 (13 to 31)	37.16	0.9744	7.310
MetaSR 3D [12]			39.59	0.9592	7.847
LISSF (Ours)			**40.47**	**0.9847**	**6.196**
Interpolation	×7.5	×7.5 (5 to 31)	28.19	0.8457	14.697
MetaSR 3D [12]			29.12	0.8592	13.839
LISSF (Ours)			**30.28**	**0.9081**	**11.838**

In addition, the quantitative metrics of the interpolation method, MetaSR3D and LISSF under another two scaling factor settings are provided in Table III. In this experiment, MetaSR3D and LISSF are the same as those in the experiments above. This experiment confirms that LISSF can achieve arbitrary scaling of spatial and spectral dimensions with only one training. LISSF can achieve stable and excellent spatial-spectral SR even with scaling factors outside the range of the training process (spatial factors larger than 4 and spectral factors larger than 5). This brings great convenience to the practical application of the model.

C. Spatial-Spectral SR Results on ARAD HS Dataset

1) Setup: In order to evaluate the generalization ability of LISSF, we perform spatial-spectral SR on the ARAD HS dataset which is not included in the training dataset. All comparison methods are exactly the same as the models in the former section and not retrained. Evaluations are also carried out under two scaling factor settings.

2) Results: The average PSNR, SSIM and SAM metrics of all methods on the ARAD HS dataset are shown in Table IV. As shown in Table IV, LISSF achieves all 6 best quantitative

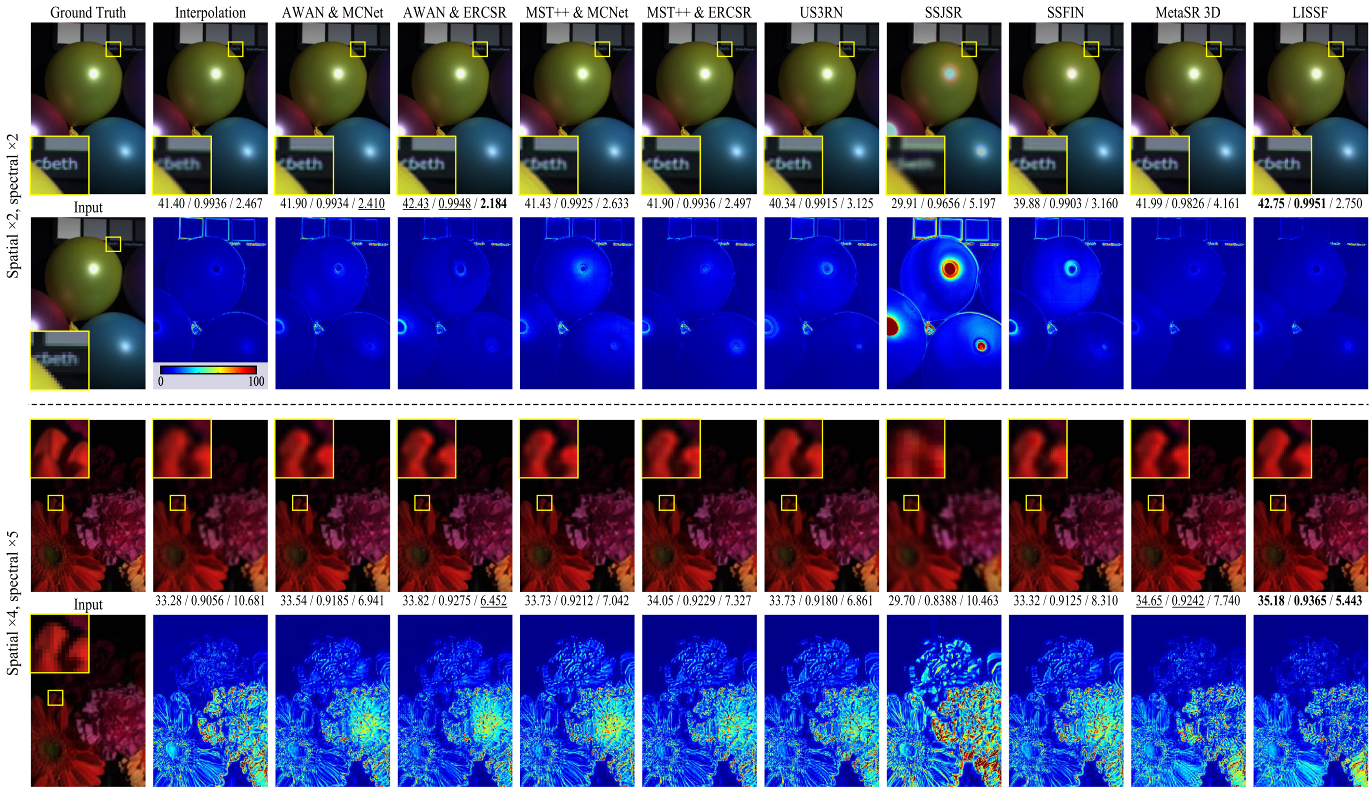

Fig. 5. Qualitative spatial-spectral SR example results of CAVE dataset (the composite images of the HSI with bands 28-13-5 as R-G-B) and the corresponding reconstruction error. The first row shows the reconstruction results of the "balloons_ms" sample with ×2 spatial scale and ×2 spectral scale (16 bands to 31 bands). The third row shows the reconstruction results of the "flowers_ms" sample with ×4 spatial scale and ×5 spectral scale (7 bands to 31 bands). The second and forth rows show the normalized reconstruction error corresponding to the first and third rows.

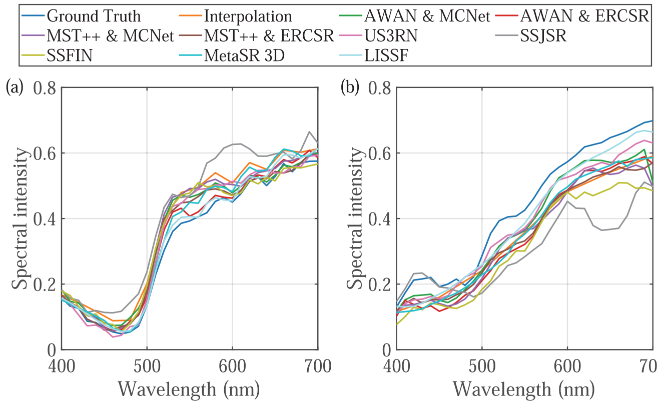

Fig. 6. Reconstrcted spectral signatures of two locations at "balloons_ms" and "flowers_ms" sample in CAVE dataset respectively.

results which is the best in all methods and MetaSR3D achieves all 6 s best quantitative results. Visual reconstruction examples of all methods under two scaling factor settings are provided in Fig. 7 where corresponding metrics are listed below the pictures and ROIs are magnified. Fig. 7 shows that LISSF achieves the richest details and the least reconstruction error. The spectral intensity of interested points are plotted in Fig. 8 where LISSF achieves the closest results to the ground truth spectrum. Interestingly, LISSF does not achieve the best performance on CAVE dataset, but it does on ARAD HS dataset. This shows that LISSF learns the deep nature of spatial-spectral SR during training, but other methods have a certain degree of overfitting on CAVE dataset. The well performance of MetaSR3D also indirectly demonstrates the effectiveness of the encoder design, because the MetaSR3D shares the same encoder structure with LISSF. It can be concluded that the proposed LISSF method have excellent generalization ability and can achieve state-of-the-art performance in spatial-spectral SR task.

D. Spatial SR Results on CAVE and ARAD HS Dataset

1) Setup: Spatial SR can be regarded as a special case of spatial-spectral SR with spectral scale of 1. We carry out a spatial SR experiments to quantitatively evaluate the performance of LISSF and other methods. Both CAVE and ARAD HS datasets are used and the spatial scaling factor is 4. The LISSF and MetaSR3D in this experiments are still the same with the models used in the experiments above, while other models are retrained for this task.

2) Results: The average PSNR, SSIM and SAM metrics of all methods are shown in Table V. SSJSR [32] achieves the worst spatial SR result among all methods, even worse than the interpolation method, the baseline of the experiments. All other DL-based methods achieve spatial SR performance no worse than the interpolation method. LISSF achieves the best performance on spatial SR of CAVE dataset, which demonstrates the effectiveness of LISSF. Besides, LISSF achieves the best

TABLE IV
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF ALL METHODS ON ARAD HS DATASET

Spatial scale	Spectral scale	Methods	PSNR↑	SSIM↑	SAM↓
×2	×2 (16 to 31)	Interpolation	40.03	0.9673	1.027
		AWAN [30] & MCNet [20]	39.30	0.9704	1.961
		AWAN [30] & ERCSR [23]	39.81	0.9760	1.877
		MST++ [31] & MCNet [20]	38.41	0.9650	2.136
		MST++ [31] & ERCSR [23]	39.66	0.9733	1.821
		US3RN [33]	37.06	0.9687	2.641
		SSJSR [32]	30.86	0.8848	4.410
		SSFIN [34]	37.20	0.9655	2.707
		MetaSR3D [12]	41.18	0.9792	1.725
		LISSF	**42.23**	**0.9822**	**1.082**
×4	×5 (7 to 31)	Interpolation	32.63	0.8773	3.131
		AWAN [30] & MCNet [20]	32.21	0.8862	4.018
		AWAN [30] & ERCSR [23]	32.76	0.8961	3.712
		MST++ [31] & MCNet [20]	32.37	0.8895	4.005
		MST++ [31] & ERCSR [23]	32.15	0.8914	4.125
		US3RN [33]	32.07	0.8888	4.080
		SSJSR [32]	28.59	0.8051	5.718
		SSFIN [34]	31.85	0.8846	4.300
		MetaSR3D [12]	34.03	0.8968	2.915
		LISSF	**34.78**	**0.9057**	**2.433**

TABLE V
QUANTITATIVE SPATIAL SR RESULTS ON CAVE AND ARAD HS DATASET

Dataset	Encoders	PSNR↑	SSIM↑	SAM↓
CAVE	Interpolation	34.71	0.9474	**3.228**
	AWAN [30] & MCNet [20]	36.05	0.9548	4.370
	AWAN [30] & ERCSR [23]	37.20	0.9631	3.373
	MST++ [31] & MCNet [20]	35.85	0.9535	4.698
	MST++ [31] & ERCSR [23]	36.79	0.9597	4.291
	US3RN [33]	36.27	0.9567	4.592
	SSJSR [32]	30.34	0.8884	8.332
	SSFIN [34]	35.08	0.9478	6.097
	MetaSR3D [12]	37.11	0.9500	5.607
	LISSF	**38.35**	**0.9643**	4.269
ARAD HS	Interpolation	32.63	0.8773	**1.146**
	AWAN [30] & MCNet [20]	34.30	0.8879	2.547
	AWAN [30] & ERCSR [23]	35.20	0.8981	2.090
	MST++ [31] & MCNet [20]	34.50	0.8852	2.455
	MST++ [31] & ERCSR [23]	35.28	0.8959	2.092
	US3RN [33]	33.83	0.8943	2.993
	SSJSR [32]	28.99	0.8087	4.977
	SSFIN [34]	32.82	0.8853	3.807
	MetaSR3D [12]	35.78	0.8990	1.421
	LISSF	**36.28**	**0.9059**	1.280

performance on spatial SR of ARAD HS dataset, which proves that LISSF has good generalization ability.

E. RGB Spectral SR Results on CAVE and ARAD HS Dataset

1) Setup: Spectral SR from RGB images can be regarded as a special case of spatial-spectral SR with spatial scale of 1 and fixed input spectral bands of 3. We conduct RGB Spectral SR experiments to quantitatively evaluate the performance of LISSF and other methods. Both CAVE and ARAD HS datasets are used and the output spectral band number is 31. As RGB images are not provided in ARAD HS, we extract the 28-13-5 bands as R-G-B in CAVE and ARAD HS datasets, which is a widely used approach in previous studies. But this creates a problem, the number of input spectral bands is different from the number we use to train LISSF and MetaSR3D in the experiments above (as few as 6 bands). Therefore, we retrained LISSF and MetaSR3D in this experiment for a fair comparison. Other methods are also retrained for this task. During training, the actual channel-wise order used is 5-13-28 to maintain the order from low wavelength to high wavelength.

2) Results: The average PSNR, SSIM and SAM metrics of all methods are shown in Table VI. As the input band number is much less than the output and three channels of RGB images are not evenly spaced, the interpolation method perform worst. Among all DL-based methods, LISSF achieves the best performance on both CAVE and ARAD HS datasets. This experiment further verifies the effectiveness and generalization ability of the LISSF method.

F. Ablation Study

In addition, we conduct an ablation study to validate the effectiveness of each proposed component.

1) Effect of encoder structure: In LISSF, the encoder is used to extract deep features from the input LR-MSI which has a significant impact on the final performance of the model. To evaluate the effectiveness of the proposed transformer-based encoder, we train LISSF with other models including VDSR [14], UNet [47], CARN [48] and RRDB [49]. All models are modified to 3D form to adapt to LISSF. The spatial-spectral SR metrics of all methods with spatial ratio of 3 and spectral ratio of 3 on CAVE dataset and ARAD HS dataset are provided in Table VII. Among all encoder choices, LISSF with the proposed transformer-based encoder achieves the best reconstruction results, which validates the effectiveness of the encoder design.

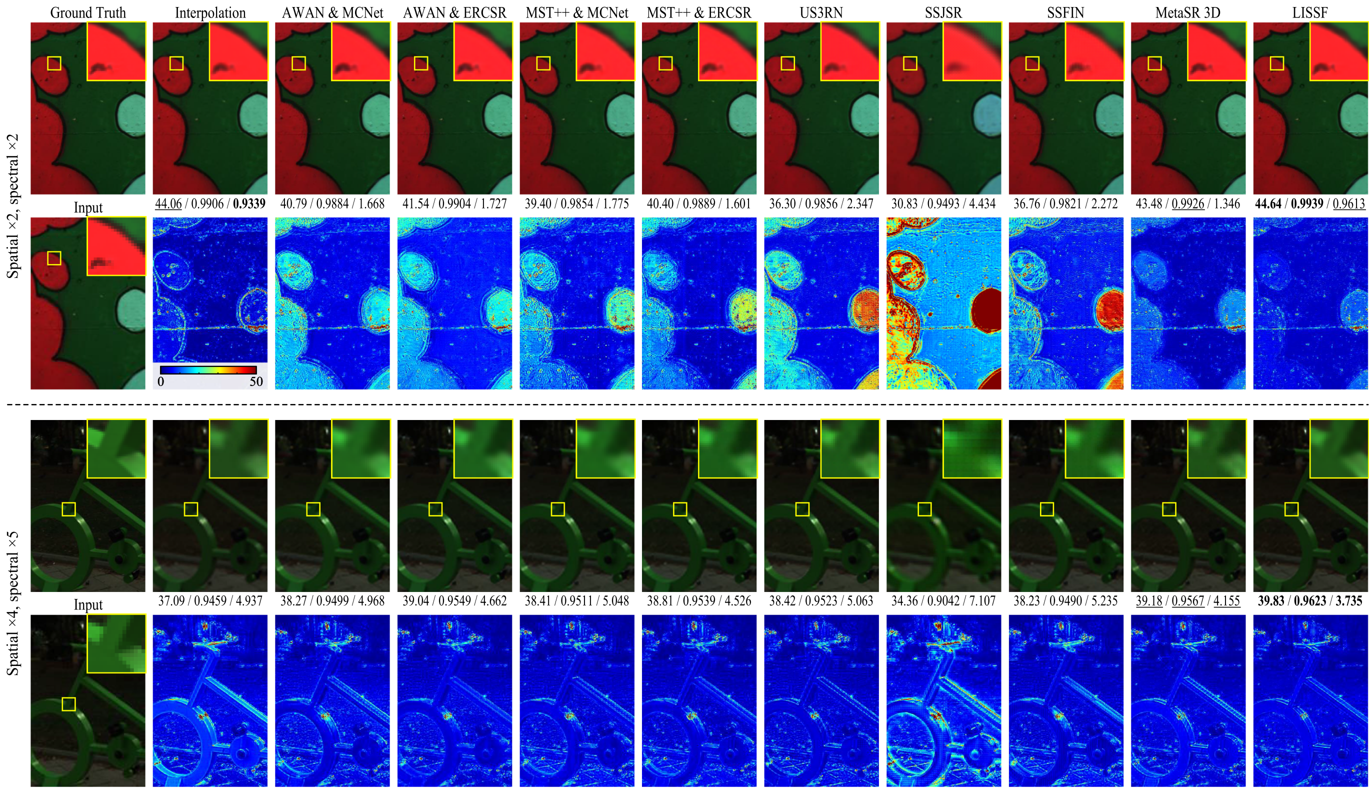

Fig. 7. Qualitative spatial-spectral SR example results of ARAD HS dataset (the composite images of the HSI with bands 28-13-5 as R-G-B) and the corresponding reconstruction error. The first row shows the reconstruction results of the "ARAD_HS_0453" sample with ×2 spatial scale and ×2 spectral scale (16 bands to 31 bands). The third row shows the reconstruction results of the "ARAD_HS_0456" sample with ×4 spatial scale and ×5 spectral scale (7 bands to 31 bands). The second and forth rows show the normalized reconstruction error corresponding to the first and third rows.

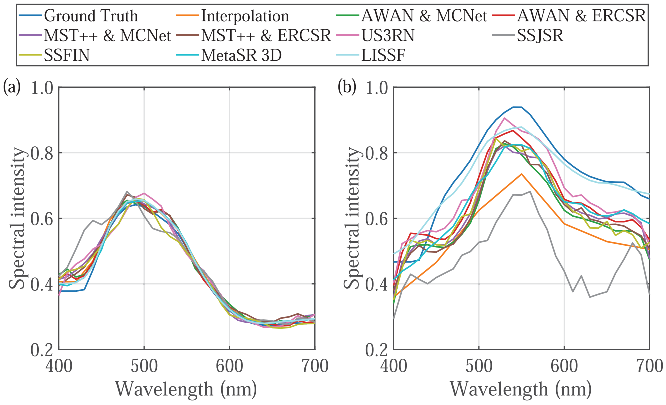

Fig. 8. Reconstrcted spectral signatures of two locations at "ARAD_HS_0453" and "ARAD_HS_0465" sample in ARAD HS dataset respectively.

TABLE VI
QUANTITATIVE RGB SPECTRAL SR RESULTS ON CAVE AND ARAD HS DATASET

Dataset	Encoders	PSNR↑	SSIM↑	SAM↓
CAVE	Interpolation	16.11	0.4327	26.04
	AWAN [30] & MCNet [20]	33.31	0.9742	9.898
	AWAN [30] & ERCSR [23]	33.38	0.9740	9.156
	MST++ [31] & MCNet [20]	33.45	0.9730	9.606
	MST++ [31] & ERCSR [23]	33.44	**0.9752**	9.483
	US3RN [33]	33.74	0.9653	10.95
	SSJSR [32]	31.83	0.9516	11.12
	SSFIN [34]	33.49	0.9719	9.775
	MetaSR3D [12]	30.00	0.7883	25.14
	LISSF	**33.82**	0.9750	**9.084**
ARAD HS	Interpolation	30.41	0.9760	6.917
	AWAN [30] & MCNet [20]	35.56	0.9856	4.475
	AWAN [30] & ERCSR [23]	35.77	0.9860	4.330
	MST++ [31] & MCNet [20]	35.75	0.9852	4.283
	MST++ [31] & ERCSR [23]	35.62	0.9836	4.548
	US3RN [33]	**36.23**	0.9888	3.922
	SSJSR [32]	35.54	0.9829	4.210
	SSFIN [34]	35.39	0.9850	4.410
	MetaSR3D [12]	35.28	0.9887	4.517
	LISSF	36.21	**0.9895**	**3.572**

2) *Effect of transformer block structure:* As the basic components of the encoder, transformer blocks have a significant impact on the performance of LISSF. In this section, we compare the performance of different transformer block variants. The MDTA3D block is replaced by MTA3D and CNN3D (3D convolutions). In MTA3D, the 3D deep-wise convolutions are replaced by standard 3D convolutions. The GDFN3D block is replaced by FN3D and CNN3D. In FN3D, the gating mechanism is removing and the 3D deep-wise convolutions are replaced by standard 3D convolutions. The spatial-spectral SR metrics of all variants with spatial ratio of 3 and spectral ratio of 3 on CAVE dataset and ARAD HS dataset are provided in Table VIII. Replacing

TABLE VII
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF METASR3D AND LISSF ON THE CAVE DATASET WITH ARBITRARY SCALING FACTORS

Dataset	Encoders	PSNR↑	SSIM↑	SAM↓
CAVE	VDSR 3D [14]	37.15	0.9729	**4.175**
	UNet 3D [47]	37.36	0.9537	6.614
	CARN 3D [48]	37.82	0.9486	7.075
	RRDB 3D [49]	38.22	0.9622	6.198
	Ours	**38.59**	**0.9781**	4.393
ARAD HS	VDSR 3D [14]	36.80	0.9322	1.669
	UNet 3D [47]	36.92	0.9357	2.131
	CARN 3D [48]	37.21	0.9392	2.249
	RRDB 3D [49]	37.48	**0.9434**	1.959
	Ours	**37.69**	**0.9434**	**1.605**

TABLE VIII
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF METASR3D AND LISSF ON THE CAVE DATASET WITH ARBITRARY SCALING FACTORS

Dataset	Transformer block	PSNR↑	SSIM↑	SAM↓
CAVE	CNN3D + GDFN3D	36.09	0.9652	5.627
	MTA3D + GDFN3D	35.91	0.9405	7.926
	MDTA3D + CNN3D	24.79	0.9291	10.180
	MDTA3D + FN3D	38.11	0.9762	**4.339**
	MDTA3D + GDFN3D	**38.59**	**0.9781**	4.393
ARAD HS	CNN3D + GDFN3D	36.61	0.9336	1.902
	MTA3D + GDFN3D	36.21	0.9311	2.358
	MDTA3D + CNN3D	35.44	0.9252	2.698
	MDTA3D + FN3D	37.54	0.9412	1.662
	MDTA3D + GDFN3D	**37.69**	**0.9434**	**1.605**

TABLE IX
QUANTITATIVE SPATIAL-SPECTRAL SR RESULTS OF METASR3D AND LISSF ON THE CAVE DATASET WITH ARBITRARY SCALING FACTORS

Dataset	Methods	PSNR↑	SSIM↑	SAM↓
CAVE	MetaSR3D [12]	37.41	0.9524	6.927
	LISSF-g	38.36	0.9663	5.792
	LISSF-l	38.34	0.9741	5.137
	LISSF-f	38.57	0.9722	5.352
	LISSF	**38.59**	**0.9781**	**4.393**
ARAD HS	MetaSR3D [12]	37.13	0.9379	2.194
	LISSF-g	37.50	0.9422	1.790
	LISSF-l	**37.74**	0.9428	1.726
	LISSF-f	37.69	**0.9439**	1.693
	LISSF	37.69	0.9434	**1.605**

MDTA3D and GDFN3D causes obvious drop of performance which demonstrates the effectiveness of the transformer block architecture.

3) Effect of decoder design: Besides the encoder, the other part that affects performance the most is the decoder, where we apply a lot of unique designs. In this section, we compare the performance of LISSF to its variants without feature unfolding, local ensemble and cell decoding. As MetaSR3D shares the same encoder structure with LISSF, it is also used as a baseline to evaluate the performance of different variants of LISSF. The spatial-spectral SR metrics of all methods under two scaling factor settings are provided in Table IX. It is clear that, comparing with the complete LISSF, LISSF-g (without grid decoding), LISSF-l (without local ensemble) and LISFF-f (without feature enhancement) all have significant performance degradations. This shows that all these components play important roles for the MLP to effectively decode deep features and perform continuous spatial-spectral reconstruction. Furthermore, we can find that not only LISSF, but LISSF-g, LISSF-l and LISFF-f also perform better than MetaSR3D. This shows that the excellent performance of LISSF is indeed due to the joint action of the encoder and decoder.

V. CONCLUSION

In this article, we propose the LISSF model which can achieve arbitrary super-resolution in both spatial and spectral dimensions. Different from spatial-spectral SR methods that learns fixed mapping from LR-MSI to HR-HSI, LISSF learns the local continuous representation of LR-MSI from discrete input independent of a specific scale. To achieve this goal, we design a transformer-based encoder and a MLP-based decoder. First, the encoder is used to transform the input LR-MSI into deep features containing both local and global information in the spatial-spectral domain. Then, the HR-HSI to be reconstructed is decomposed into individual coordinates for processing, and a feature vector is generated for each coordinate. At last, the decoder is applied to project the feature vectors to intensity values at specific spatial-spectral coordinates. Detailed comparisons and ablation studies were carried out to validate the effectiveness of LISSF. Experiments shows that LISSF can achieve better spatial-spectral SR results with arbitrary scales than state-of-the-art methods retrained for a specific scale, which has significant convenience in practical applications.

REFERENCES

[1] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, “Imaging spectrometry for earth remote sensing,” *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[3] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[4] G. Lu and B. Fei, “Medical hyperspectral imaging: A review,” *J. Biomed. Opt.*, vol. 19, no. 1, 2014, Art. no. 010901.

[5] H. Akbari, Y. Kosugi, K. Kojima, and N. Tanaka, “Detection and analysis of the intestinal ischemia using visible and invisible hyperspectral imaging,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 8, pp. 2011–2017, Aug. 2010.

[6] X. Wei, W. Li, M. Zhang, and Q. Li, “Medical hyperspectral image classification based on end-to-end fusion deep neural network,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 11, pp. 4481–4492, Nov. 2019.

[7] A. A. Gowen, C. P. O’Donnell, P. J. Cullen, G. Downey, and J. M. Frias, “Hyperspectral imaging–an emerging process analytical tool for food quality and safety control,” *Trends Food Sci. Technol.*, vol. 18, no. 12, pp. 590–598, 2007.

[8] J.-H. Cheng and D.-W. Sun, “Recent applications of spectroscopic and hyperspectral imaging techniques with chemometric analysis for rapid inspection of microbial spoilage in muscle foods,” *Comprehensive Rev. Food Sci. Food Saf.*, vol. 14, no. 4, pp. 478–490, 2015.

[9] A. Picon, A. Vicente, S. Rodriguez-Vaamonde, J. Armentia, J. A. Arteche, and I. Macaya, “Ladle furnace slag characterization through hyperspectral reflectance regression model for secondary metallurgy process optimization,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3506–3512, Aug. 2018.

[10] B. Arad et al., “NTIRE 2018 challenge on spectral reconstruction from RGB images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1042–104209.

[11] B. Arad, R. Timofte, O. Ben-Shahar, Y.-T. Lin, and G. D. Finlayson, “NTIRE 2020 challenge on spectral reconstruction from an RGB image,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 446–447.

[12] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, “Meta-SR: A magnification-arbitrary network for super-resolution,” in *Proc. IEEE/CVF Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1575–1584.

[13] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[14] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.

[15] C. Ledig et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proc. IEEE 30th Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[16] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1132–1140.

[17] Y. Yuan, X. Zheng, and X. Lu, “Hyperspectral image superresolution by transfer learning,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1963–1974, May 2017.

[18] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, “Hyperspectral image spatial super-resolution via 3D full convolutional neural network,” *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1139.

[19] J. Li et al., “Hyperspectral image super-resolution by band attention through adversarial learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4304–4318, Jun. 2020.

[20] Q. Li, Q. Wang, and X. Li, “Mixed 2D/3D convolutional network for hyperspectral image super-resolution,” *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1660.

[21] J. Jiang, H. Sun, X. Liu, and J. Ma, “Learning spatial-spectral prior for super-resolution of hyperspectral imagery,” *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1082–1096, 2020.

[22] D. Liu, J. Li, and Q. Yuan, “A spectral grouping and attention-driven residual dense network for hyperspectral image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7711–7725, Sep. 2021.

[23] Q. Li, Q. Wang, and X. Li, “Exploring the relationship between 2D/3D convolution for hyperspectral image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8693–8703, Oct. 2021.

[24] X. Han, J. Yu, J. Luo, and W. Sun, “Reconstruction from multispectral to hyperspectral image using spectral library-based dictionary learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1325–1335, Mar. 2019.

[25] C. Yi, Y.-Q. Zhao, and J. C.-W. Chan, “Spectral super-resolution for multispectral image based on spectral improvement strategy and spatial preservation strategy,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9010–9024, Nov. 2019.

[26] U. B. Gewali, S. T. Monteiro, and E. Saber, “Spectral super-resolution with optimized bands,” *Remote Sens.*, vol. 11, no. 14, pp. 1–24, 2019.

[27] P. V. Arun, K. M. Buddhiraju, A. Porwal, and J. Chanussot, “CNN based spectral super-resolution of remote sensing images,” *Signal Process.*, vol. 169, 2020, Art. no. 107394.

[28] X. Zheng, W. Chen, and X. Lu, “Spectral super-resolution of multispectral images using spatial–spectral residual attention network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5404114.

[29] Z. Shi, C. Chen, Z. Xiong, D. Liu, and F. Wu, “HSCNN+: Advanced CNN-based hyperspectral recovery from RGB images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 939–947.

[30] J. Li, C. Wu, R. Song, Y. Li, and F. Liu, “Adaptive weighted attention network with camera spectral sensitivity prior for spectral reconstruction from RGB images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 462–463.

[31] Y. Cai et al., “MST++: Multi-stage spectral-wise transformer for efficient spectral reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 745–755.

[32] S. Mei, R. Jiang, X. Li, and Q. Du, “Spatial and spectral joint super-resolution using convolutional neural network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4590–4603, Jul. 2020.

[33] Q. Ma, J. Jiang, X. Liu, and J. Ma, “Deep unfolding network for spatiospectral image super-resolution,” *IEEE Trans. Comput. Imag.*, vol. 8, pp. 28–40, 2022.

[34] Q. Ma, J. Jiang, X. Liu, and J. Ma, “Multi-task interaction learning for spatiospectral image super-resolution,” *IEEE Trans. Image Process.*, vol. 31, pp. 2950–2961, 2022.

[35] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser, “Learning shape templates with structured implicit functions,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7154–7164.

[36] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, “Local deep implicit functions for 3D shape,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4857–4866.

[37] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, “Scene representation networks: Continuous 3d-structure-aware neural scene representations,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019,, vol. 32.

[38] C. Jiang et al., “Local implicit grid representations for 3D scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6001–6010.

[39] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[40] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7462–7473.

[41] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, “MetaSDF: Meta-learning signed distance functions,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 10136–10147.

[42] Y. Chen, S. Liu, and X. Wang, “Learning continuous image representation with local implicit image function,” in *Proc. IEEE/CVF Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8624–8634.

[43] R. Xu, M. Yao, C. Chen, L. Wang, and Z. Xiong, “Continuous spectral reconstruction from RGB Images via implicit neural representation,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 78–94.

[44] K. Zhang, “Implicit neural representation learning for hyperspectral image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5500212.

[45] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5728–5739.

[46] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, “Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum,” *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.

[47] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Lecture Notes Comput. Sci. Including Subseries Lecture Notes Artif. Intell. Lecture Notes Bioinf.*, 2015, vol. 9351, pp. 234–241.

[48] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 252–268.

[49] X. Wang et al., “ESRGAN: Enhanced super-resolution generative adversarial networks,” in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.