A Deep Estimation-Enhancement Unfolding Framework for Hyperspectral Image Reconstruction

Zhen Fang $^1$, Xu Ma $^2$, and Gonzalo Arce $^2$

$^1$Beijing Institute of Technolog
$^2$Affiliation not available

Abstract

Coded aperture snapshot spectral imager (CASSI) can recover three-dimensional hyperspectral images (HSIs) from two-dimensional compressive measurements. Recently, deep unfolding approaches were shown impressive reconstruction performance among various algorithms. Existing deep unfolding methods usually employ linear projection methods to guide the iterative learning process. However, the linear projections do not include trainable parameters and ignore the essential characteristics of HSI. This paper proposes a novel learning-based deep estimation-enhancement unfolding (DEEU) framework to improve the HSI reconstruction. The deep estimation-enhancement (DEE) module is used to guide the iterative learning process of the network based on the prior information of the CASSI system, and then exploits the intrinsic features of the reconstructed HSI in both spectral and spatial dimensions. In addition, a multi-prior ensemble learning module is proposed to further improve the reconstruction performance without increasing runtime. As with most of deep unfolding methods, we plug a convolutional neural network as a denoiser in each stage of the DEEU framework, which finally forms the proposed DEEU-Net. Comprehensive experiments on both simulation and real datasets demonstrate that the effectiveness of our DEEU framework, and our DEEU-Net can achieve both high reconstruction quality and speed, outperforming the state-of-the-art methods.
A Deep Estimation-Enhancement Unfolding Framework for Hyperspectral Image Reconstruction

Zhen Fang, Xu Ma, Senior Member, IEEE, and Gonzalo R. Arce, Life Fellow, IEEE

Abstract—Coded aperture snapshot spectral imager (CASSI) can recover three-dimensional hyperspectral images (HSIs) from two-dimensional compressive measurements. Recently, deep unfolding approaches were shown impressive reconstruction performance among various algorithms. Existing deep unfolding methods usually employ linear projection methods to guide the iterative learning process. However, the linear projections do not include trainable parameters and ignore the essential characteristics of HSI. This paper proposes a novel learning-based deep estimation-enhancement unfolding (DEEU) framework to improve the HSI reconstruction. The deep estimation-enhancement (DEE) module is used to guide the iterative learning process of the network based on the prior information of the CASSI system, and then exploits the intrinsic features of the reconstructed HSI in both spectral and spatial dimensions. In addition, a multi-prior ensemble learning module is proposed to further improve the reconstruction performance without increasing runtime. As with most of deep unfolding methods, we plug a convolutional neural network as a denoiser in each stage of the DEEU framework, which finally forms the proposed DEEU-Net. Comprehensive experiments on both simulation and real datasets demonstrate that the effectiveness of our DEEU framework, and our DEEU-Net can achieve both high reconstruction quality and speed, outperforming the state-of-the-art methods.

Index Terms—Computational imaging, compressive spectral imaging, deep learning, deep unfolding.

I. INTRODUCTION

Hyperspectral imaging (HSI) captures a number of narrow spectral images within a continuous spectral range, and constructs a three-dimensional (3D) spatio-spectral data cube of the target scene. HSIs provide much richer and more detailed spectral information than RGB images, and have been widely used in object detection [1], image recognition [2], remote sensing [3] and other fields. Conventional HSI systems obtain the 3D spatio-spectral data cube by scanning the target scene along the spectral or spatial directions, which is not suitable to capture the dynamic scenes [4]–[6]. Thus, various compressive spectral imaging systems have been proposed to accelerate the data acquisition process [6]–[12]. Recently, coded aperture snapshot spectral imager (CASSI) has achieved remarkable performance, which acquires 3D HSI cube with only a single or a few two-dimensional (2D) compressive measurements [4], [6].

As shown in Fig. 1, the HSI cube of a target scene is first modulated by a block/unblock coded aperture in the spatial coordinates. Then, different spectral channels of the coded HSI cube are shifted to different spatial locations by a dispersive prism. Finally, the modulated HSI is projected onto a 2D detector. In the CASSI system, all HSI voxels at each spatial location (green boxes in F) are dispersed across multiple detector pixels. Each detector pixel will receive the information of multiple HSI voxels from different spectral channels (red boxes in F). The most important task of the CASSI system is to quickly and faithfully reconstruct the HSI cube from the compressive measurements.

Model-based iterative methods were first proposed for HSI reconstruction, which mainly rely on various hand-crafted priors, such as the total variation [13], sparsity [12], [14] and low-rank [15] regularizations. These methods have strong interpretability, but are time-consuming and have inferior reconstruction quality and generalization ability. Recently, due to the success of deep learning approaches in many fields [16]–[20], learning-based methods were developed for HSI reconstruction and exhibited excellent performance. At first, several end-to-end methods were proposed that employ various powerful networks to learn the brute-force mapping functions between the compressive measurements and HSI cube. However, the end-to-end models often lack interpretability, since they ignore the underlying imaging principles in CASSI system [20]–[22].

Deep unfolding (DU) methods take advantages from both model-based methods and end-to-end methods, thus rendering powerful deep learning networks with physical interpretability. DU methods adopt multi-stage networks, each stage of which usually includes two steps, namely a projection-update step and a denoising step [23]–[25]. The existing DU methods for HSI reconstruction employ linear projection methods in the projection-update step to guide the iterative learning process, but they replace the traditional hand-craft priors with some deep priors in the denoising step. With the superiority of deep priors, DU methods obtain excellent reconstruction performance with a few stages.

However, the linear projection methods have two drawbacks. First, they update the current estimation result based on only a single compressive measurement and the system sensing matrix, which does not include any trainable parameters. Second, the linear projection methods are usually formulated as quadratic regularized least-squares problems that are based on mathematical analysis and implemented by a few fixed matrix operations. Thus, they fall short to consider the highly correlated characteristics of HSI data cube along the spatial
and spectral dimensions. Therefore, the existing DU methods mainly focus on the development of denoising networks [23]. Although DU methods proposed to date have their own advantages in different aspects, they cannot achieve high reconstruction quality and fast running speed simultaneously [19].

To address these problems, this paper proposes a novel learning-based DU framework, namely deep estimation-enhancement unfolding (DEEU) framework for the HSI reconstruction. Different from the existing DU methods, the DEEU framework replaces the linear projection with a deep estimation-enhancement (DEE) module to guide the iterative learning process. DEE module consists of a deep projection-estimation (DPE) block and a spatio-spectral feature enhancement (SSFE) block. As shown in Fig. 1, as the HSI cube includes a number of spectral channels, establishing the mapping function between a single compressive measurement and the reconstructed HSI cube becomes a non-local estimation task. However, based on the spatio-spectral correlation of the projection imaging process in CASSI system, the DPE block uses a few 2D convolution layers to extract the local image feature from specific detector pixels and then update the corresponding HSI voxel values. This method will prove to be an efficient local estimation. Subsequently, the SSFE block is used to further explore the intrinsic features of the estimation result of the DPE block in both spectral and spatial dimensions, which means the spectral correlation and the spatial similarity.

Inspired by ensemble learning methods recently introduced [19], [26], [27], the proposed DEEU framework employs a multi-prior ensemble learning (MPEL) module to connect previous stages and latter stages of the network. The MPEL module can accelerate the network training process and further improve the reconstruction quality without additional training cost and inference time. In addition, since both the DEE module and the MPEL module adopt simple 2D convolutional neural networks (CNNs), the DEEU framework exploits the power of deep learning technologies and preserves fast processing capability.

Due to the advantage of CNN in computational efficiency [19], a 2D CNN-based denoiser is plugged in each stage of the DEEU framework, which forms an iterative network, dubbed DEEU-Net, to quickly and faithfully reconstruct the HSI cube. It is worth noting that existing CNN-based DU methods achieve fast reconstruction speed but inferior reconstruction quality [6], [19], [24], [25]. Moreover, the transformer-based DU methods can improve the reconstruction quality, but the training cost and inference time will significantly increase [20], [22], [23]. Thus, existing DU methods cannot achieve real-time and high-quality reconstruction for dynamic scenes [23]–[25]. On the other hand, the proposed DEEU-Net can effectively improve the reconstruction quality and computational efficiency simultaneously. The main contributions of this paper are summarized as follows:

- This paper first proposes the novel learning-based DEEU framework, which first uses the DEE module to replace the linear projection methods and guide the iterative learning process. The DEE module can efficiently update each HSI voxel in a local estimation process, and then explore the spectral correlation and the spatial similarity of HSI cube.
- The MPEL module is then proposed in the DEEU framework to connect the previous stages and latter stages, which can improve the reconstruction quality without additional training cost and inference time.
- The 2D CNN-based denoiser is applied in each stage of the proposed DEEU-Net to preserve computation speed.
- Experimental results demonstrate the effectiveness and generalization ability of our DEEU framework compared to traditional DU methods employing linear projection. Moreover, the proposed DEEU-Net shows significant advantages in reconstruction quality and computational efficiency over the existing state-of-the-art (SOTA) methods for HSI reconstruction.

II. HSI RECONSTRUCTION IN CASSI SYSTEM

A. Mathematical model of CASSI system

As shown in Fig. 1, let $F \in \mathbb{R}^{N_x \times N_y \times N_z}$ represent the 3D HSI cube of the target scene, where $N_z$, $N_y$ are the spatial coordinates and $N_z$ is the spectral coordinate. Let $C \in \mathbb{R}^{N_x \times N_y}$ represent the coded aperture. In the CASSI system, $F$ is first modulated by the coded aperture in the spatial domain, which can be formulated as

$$F_1(n_x, n_y, n_\lambda) = F(n_x, n_y, n_\lambda)C(n_x, n_y),$$  \hspace{1cm} (1)
where $F_1 \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ represents the coded data cube, where $n_x = 1, ..., N_x$ and $n_y = 1, ..., N_y$ index its spatial coordinates, and $n_\lambda = 1, ..., N_\lambda$ indexes its spectral channel. Then, different spectral channels of $F_1$ are shifted along the vertical direction by the dispersive prism. This process can be described by a disperive function $\mathfrak{D}()$ as following

$$F_2[n_x, n_y, n_\lambda] = \mathfrak{D}(F_1[n_x, n_y, n_\lambda]) = \mathfrak{D}(F_1)[n_x, n_y + \alpha \cdot (n_\lambda - 1), n_\lambda] = \mathfrak{D}(F_1)[n_x, n_y, n_\lambda + \alpha \cdot (n_\lambda - 1), n_\lambda],$$  

(2)

where $F_2 \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ represents the shifted data cube, $N_u = [N_y + \alpha \cdot (N_\lambda - 1)]$, and $\alpha$ is the dispersion rate of the dispersive prism. Moreover, we can define an inverse transformation function of $\mathfrak{D}(\cdot)$ as $\mathfrak{D}^{-1}(\cdot)$, which can be written as

$$F_1(n_x, n_y, n_\lambda) = \mathfrak{D}^{-1}(F_2)[n_x, n_y + \alpha \cdot (n_\lambda - 1), n_\lambda].$$  

(3)

Then, $F_2$ is integrated through the spectral domain, and forms a single compressive measurement $Y \in \mathbb{R}^{N_x \times N_u}$ on the 2D focal plane array detector, which is formulated as

$$Y(n_x, n_u) = \sum_{n_\lambda=1}^{N_\lambda} F_2(n_x, n_u, n_\lambda) = \sum_{n_\lambda=1}^{N_\lambda} \left\{ F[n_x, n_u - \alpha \cdot (n_\lambda - 1), n_\lambda] \cdot O[n_x, n_u] + O[n_x, n_u] \right\}.$$  

(4)

where $n_u = 1, ..., N_u$ indexes the vertical coordinate in the detector plane, $O \in \mathbb{R}^{N_x \times N_u}$ represents the additive sensing noise on the detector. Furthermore, Eq. (4) can be rewritten in the matrix form:

$$Y = \Phi f + o,$$  

(5)

where $y$, $f$ and $o$ are the vectorized representations of $Y$, $F$ and $O$, respectively; $\Phi \in \mathbb{R}^{N_x \times N_u N_\lambda N_\lambda}$ is the sensing matrix representing the forward imaging model of CASSI system, which includes the effects of the coded aperture and dispersive prism. According to Ref. [8], $\Phi$ is a sparse, fat and highly structured matrix. That is, most of the elements in $\Phi$ are zero, and $\Phi$ has more columns than rows.

The well-designed CASSI system guarantees the fast acquisition of compressive measurements. The next important task is how to reconstruct the underlying 3D HSI cube from the compressive measurements quickly and faithfully.

**B. Conventional DU method for HSI reconstruction**

The existing HSI reconstruction methods aim at solving the following optimization problem

$$\hat{f} = \arg\min_{f} \frac{1}{2} \|y - \Phi f\|^2 + \gamma \Omega(f),$$  

(6)

where $\frac{1}{2} \|y - \Phi f\|^2$ is the data fidelity term, $\Omega(f)$ is an image prior to confine the solution space, and $\gamma$ is a weight parameter. According to Refs. [6] and [23], with the help of the auxiliary variable $v$ and weighting coefficient $\mu$, Eq. (6) can be reformulated as

$$\begin{align*}
\hat{f}, \hat{v} = \arg\min_{f,v} & \frac{1}{2} \|y - \Phi f\|^2 + \gamma \Omega(f) + \frac{\mu}{2} \|v - f\|^2.
\end{align*}$$  

(7)

Specifically, for the $k$th iteration in the DU method, Eq. (7) can be decomposed into two iterative sub-steps, i.e., the projection-update step $P_k(\cdot)$ and the denoising step $D_k(\cdot)$, for where $k = 1, ..., K$ and $K$ is the total number of iterations (stages). Then, $P_k(\cdot)$ and $D_k(\cdot)$ are used to solve for $f^k$ and $v^k$ in the $k$th stage, respectively, which are described as follows

**Projection-update step:** based on the denoising prior $v^{k-1}$ in the $(k - 1)$th stage, the sensing matrix of CASSI system and a single compressive measurement, $f^k$ is first updated as:

$$f^k = P_k(y, \Phi, v^{k-1}).$$  

(8)

**Denoising step:** a trainable neural network is used to map the estimation result of $P_k(\cdot)$ to the desired HSI $[6]$. That is,

$$v^k = \arg\min_{v} \frac{\mu}{2} \|v - f^k\|^2 + \gamma \Omega(v) = D_k(\theta_k, f^k),$$  

(9)

where $\theta_k$ represents the network parameters of the denoiser in the $k$th stage.

According to Ref. [6], [23] and [24], the deep denoising prior showed significant advantage over the traditional hand-craft priors, and thus the DU method can achieve superior reconstruction performance with a few stages.

In contrast to model-based reconstruction methods, the existing DU methods employ linear projection operations in the projection-update step, but replace the hand-craft priors with different deep denoising priors. Based on different model-based methods and neural networks, various DU methods for HSI reconstruction have been proposed [6], [23], [24].

For simplicity, we use the generalized alternating projection (GAP) framework [24] as an example to illustrate the calculation process of linear projection methods. In the GAP framework, Eq. (8) can be solved as

$$f^k = v^{k-1} + \Phi^T (\Phi \Phi^T)^{-1} (y - \Phi v^{k-1}),$$  

(10)

where $(\Phi \Phi^T)^{-1}$ is pre-calculated and stored in advance. Equation (10) is implemented by a few matrix operations, which is essentially a gradient descent process and does not include any learnable parameters. Moreover, Eq. (10) falls short to consider the spectral and spatial characteristics of the HSI cube, which will affect the reconstruction performance adversely [22], [28], [29]. When the desired HSI cube consists of a number of spectral channels, linear projection methods may limit the generalization ability of the DU methods. Thus, most of existing DU methods rely heavily on the deep learning denoiser to improve the reconstruction quality, which is defined in Eq. (9). However, these DU methods cannot improve the reconstruction quality and computing speed simultaneously [19].

**C. Spatio-spectral correlation of CASSI projection**

In this section, we present a detailed discussion on the spatio-spectral correlation of CASSI projections. As described in Section 2.1, due to the modulation effect of the dispersive prism, different spectral channels of the coded HSI cube are shifted to different spatial location as defined in Eq. (2). In Fig.
1. if the underlying HSI cube consists of $N_\lambda$ spectral channels, the spatio-spectral correlation of the CASSI projection can be expressed as follows. Each spatial position in the HSI cube $F$ is associated with $N_\lambda$ voxels, which belong to different spectral channels. After dispersion, these voxels will be sensed by $N_\lambda$ different detector pixels. Moreover, these $N_\lambda$ detector pixels span a long distance in the spatial domain, where the span distance can be calculated as $ds = \alpha \cdot (N_\lambda - 1)$, which is defined in Fig. 1 and Eq. (2). At the same time, each detector pixel may sense at most $N_\lambda$ HSI voxels, which belong to different spatial and spectral coordinates in the original HSI cube. Moreover, these HSI voxels also span a long distance in the original HSI cube $F$, and the span distance is also equal to $ds$.

Based on the above discussion, there will be some drawbacks if we directly use a neural network to process the compressive measurement and obtain the estimation result. First, according to Eq. (5), the sensing matrix $\Phi$ is a large and sparse matrix. Thus, when we use fully connected layers to learn the mapping function between a single compressive measurement and the HSI cube, it requires a huge number of network parameters. Second, when the HSI cube consists of a number of spectral channels, the span distance $ds$ is large, which results in a non-local estimation process. If a CNN is used to implement the non-local estimation based on a single compressive measurement, we need to stack multiple convolution layers to increase the sensing range of the network. This kind of approach is also used in many end-to-end methods. However, the brute-force mapping requires the network to extract the HSI features from a large amount of redundant information, thus ignores the system-structure prior and lacks interpretability. Finally, the prohibitive computational burden is also unacceptable if the popular self-attention mechanism was used to capture the non-local dependencies.

Thus, the desired reconstruction network should be carefully designed based on the spatio-spectral correlation of the CASSI projection process. A reasonable strategy is to estimate each HSI voxel from the detector pixels that can sense it. In this case, the above non-local estimation task can be solved by a local estimation method.

III. DEEP ESTIMATION-ENHANCEMENT UNFOLDING FRAMEWORK

To overcome the limitations of the existing DU frameworks, we propose a learning-based DEEU framework for HSI reconstruction. The architecture of the DEEU framework is shown in Fig. 2. Based on the spatio-spectral correlation of CASSI projection and the characteristics of HSI cube, the DEEU framework first employs a learning-based DEE module in the projection-update step to guide the iterative learning process. Then, the CNN-based MPEL module is used to connect the previous stages and latter stages, thus further improving the reconstruction quality without additional training cost and inference time.

A. Deep estimation-enhancement module

According to Eq. (10), traditional linear projection methods have less degree of optimization freedom to fulfill the projection-update step. To circumvent this problem, the learning-based DEE module is developed to replace the linear projection. As shown in Fig. 3, the DEE module is cascaded by two blocks, namely DPE block and SSFE block. The former block updates the estimation result, and the latter one exploits and enhances the spectral correlation and spatial similarity of the estimation result. Next, we will describe the two blocks in details.

1) Deep projection estimation block: In this section, a CNN-based DPE block is proposed based on the spatio-spectral correlation of CASSI projection. The DPE block can extract local image features from the specific detector pixels and then update the corresponding HSI voxels.

Figure 3(a) shows the architecture of DPE block. For the DPE block in the $k$th stage, we first calculate the projection error, denoted as $P_{e}^{k-1} \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$, between the denoising prior of the $(k-1)$th stage and the actual compressive measurement. This is formulated as

$$P_{e}^{k-1} = F_{Re}[abs(y - \Phi v^{k-1})]$$

(11)

where $F_{Re}()$ is a reshaping function and $abs(\cdot)$ is the absolute value function. In the first stage, $P_{e}^{0} = Y$ and $v^{0} = 0$. In addition, based on the spatio-spectral correlation of CASSI projection, we construct a dispersion matrix $DM \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ to indicate the relation between each detector pixel and the corresponding HSI voxels:

$$DM = \mathcal{Z}(dm)/T,$$

(12)

where $\mathcal{Z}(\cdot)$ was defined in Eq. (2), $dm \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ is initialized to a one-valued matrix and $T \in \mathbb{R}^{N_x \times N_y}$ records the number of non-zero elements at each spatial location of $\mathcal{Z}(dm)$.

By the aid of $DM$, the DPE block extracts image features from each local region of $P_{e}^{k-1}$, and then updates the corresponding HSI voxels. Considering the advantages of convolution layers in extracting the local image features, we use a CNN-based $ConvBlock$ to process $P_{e}^{k-1}$:

$$f_{ConvBlock}^{k} = ConvBlock(P_{e}^{k-1} \odot DM),$$

(13)

where $f_{ConvBlock}^{k} \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ has the same representation with $F_2$, which is the shifted representation of $F$, and $\odot$ denotes the element-wise product. As shown in Fig. 3(a), the $ConvBlock$ has a simple network structure consisting of three 2D convolution layers. Then, in order to obtain the new estimation result of $F$ with $f_{ConvBlock}^{k}$ and $v^{k-1}$, we need to calculate the shifted representation of $v^{k-1}$, that is,

$$v_{disp}^{k-1} = \mathcal{Z}(v^{k-1}).$$

(14)

Finally, according to Eqs. (13) and (14), the output of DPE block in the $k$th stage is defined as

$$z^{k} = DPE^{k}(P_{e}^{k-1}, DM, v^{k-1}) = \mathcal{Z}[Relu(f_{ConvBlock}^{k} + v_{disp}^{k-1})]^T,$$

(15)

where $DPE^{k}(\cdot)$ represents the mapping function in the $k$th stage; $Relu(\cdot)$ is the rectified linear unit (ReLU) function; $\mathcal{Z}(\cdot)$ was defined in Eq.(3). Since the intensities of HSI voxels are always non-negative, $Relu(\cdot)$ removes the negative elements.
Fig. 2. Schematic diagram of the DEEU framework with $K$ stages. $Y$ and $\Phi$ are a single compressive measurement and the sensing matrix of CASSI, respectively. $P_k(\cdot)$ and $D_k(\cdot)$ represent the projection-update step and denoising step in the $k$th stage, respectively. $P_k(\cdot)$ is realized by the proposed learning-based DEE module, and $D_k(\cdot)$ is realized by a CNN-based denoiser. The CNN-based MPEL module is used to fuse the denoising priors from previous three adjacent stages, which serve as the input of the DEE module.

Fig. 3. The architecture of DEE module in the $k$th stage. The DEE module is cascaded by (a) the DPE block and (b) the SSFE block.

in the estimation result $z^k$. Unlike the brute-force mapping and non-local estimation functions, the DPE block can directly and efficiently learn the mapping between a single compressive measurement and the HSI cube in a local estimation process.

2) Spatio-spectral feature enhancement block: According to Eq. (10), linear projection methods do not take into account the spectral and spatial features, which is the spectral correlation and the spatial similarity of the HSI cube [6]. It is observed that different spectral channels of HSI, especially the adjacent spectral channels have similar spatial structures [4], [6].

It is noted that the HSI voxels sensed by one detector pixel come from different spectral channels, and those voxels may be far apart from each other in the original HSI cube. Thus, the spectral correlation and the spatial similarity of the output of the DPE block need to be further exploited. To address this challenge, we propose a SSFE block to further process the estimation result of DPE block.

As shown in Fig. 3(b), the SSFE block includes a spectral feature enhancement block (FEB) and a spatial FEB, which can exploit the spectral correlation and the spatial similarity of the input, respectively. For the spectral FEB in the $k$th stage, we first use a $1 \times 1$ convolution layer to process $z^k$ and obtain $f^{k}_{\text{spe} \text{gap}}$. Then, we apply the spectral channel attention on $f^{k}_{\text{spe} \text{gap}}$. That is, we use the global average pooling function to record the global spectral characteristic of each spectral channel, which is defined as $f^{k}_{\text{gap}}$. After that, two fully connected layers followed by the ReLU and sigmoid functions are utilized to encode-decode $f^{k}_{\text{gap}}$, and the weight factor $f^{k}_{\text{w}f}$ is obtained. These operations can be described as

$$
\begin{align*}
    f^{k}_{1 \times 1} &= F^{1 \times 1}_{\text{Conv}}(z^k), \\
    f^{k}_{\text{gap}} &= \text{gap}(f^{k}_{1 \times 1}), \\
    f^{k}_{\text{w}f} &= \text{Sigmod}(F^{2 \times 2}_{\text{FC}}(\text{Relu}(F^{2 \times 2}_{\text{FC}}(f^{k}_{\text{gap}}))))
\end{align*}
$$

(16)

where $F^{1 \times 1}_{\text{Conv}}(\cdot)$ represents the $1 \times 1$ convolution function; $\text{gap}(\cdot)$ and $\text{Sigmod}(\cdot)$ represent the global average pooling and the sigmoid functions, respectively; $F^{2 \times 2}_{\text{FC}}(\cdot)$ and $F^{2 \times 2}_{\text{FC}}(\cdot)$ represent the two fully connected layers. Finally, we impose $f^{k}_{\text{w}f}$ on $f^{k}_{1 \times 1}$, and add the skip connection between the input and output of the spectral FEB. Thus, the output of spectral FEB in the $k$th stage can be formulated as

$$
\begin{align*}
    f^{k}_{\text{spe}} &= \text{SpeFEB}^k(z^k) = z^k + f^{k}_{\text{w}f} \odot f^{k}_{1 \times 1},
\end{align*}
$$

(17)

where $\text{SpeFEB}^k(\cdot)$ represents the mapping function in the $k$th stage, which learns the nonlinear spectral correlation between different spectral channels.

Then, we use a spatial FEB to exploit the spatial similarity of $f^{k}_{\text{spe}}$ in the spatial dimension. Consider that each spectral channel of HSI cube is a natural image corresponding to the same target scene. Then, all spectral channels should vary similarly in the spatial domain, and have similar texture structure. In addition, due to the advantages of CNNs in terms of the local connection and shift-invariance, we implement a ResBlock [30] in the spatial FEB, which can extract the spatial
features of $f_{spe}^k$ efficiently. The output of spatial FEB in the $k^{th}$ stage is given by

$$f_{spe}^k = \text{SpaFEB}(f_{spe}^k) = F_{\text{Conv}}^{1\times1}[\text{ResBlock}(f_{spe}^k)],$$

where $\text{SpaFEB}(\cdot)$ represents the mapping function, $\text{ResBlock}(\cdot)$ represents a ResBlock, whose architecture is shown in Fig. 3(b). Combining the spectral FEB and spatial FEB together, the output of SSFE block in the $k^{th}$ stage is finally expressed as

$$f^k = \text{SSFEm}(z^k) = \text{Relu}(\{z^k + \text{SpaFEB}(\text{SpaFEB}(z^k))\}),$$

where $\text{SSFEm}(\cdot)$ represents the mapping function, and $\text{Relu}(\cdot)$ enforces each voxel of $f^k$ to be non-negative.

### B. Multi-prior ensemble learning module

In the existing DU methods, the denoisers in different stages play different roles in the iterative learning process [19]. As the number of stages increases, the DU methods become difficult to train, and the reconstruction quality is not improved obviously [23]. That is because the data transmission between the previous stages and latter stages becomes inefficient as the depth of network increases. Specifically, in the forward propagation, it is difficult for the latter stages to utilize the image features from the previous stages. Meanwhile, in the back-propagation process the gradient information from the latter stages is difficult to reach the previous stages.

![Multi-Prior Ensemble Learning Module](Image)

**Fig. 4.** The architecture of MPEL module in the $(k+3)^{th}$ stage.

Ensemble learning can exploit the complementary advantages of multiple priors, and has achieved SOTA results in various tasks [19], [26], [27]. Thus, as shown in Figs. 2 and 4, we develop an MPEL module to connect the previous stages and latter stages in the DEEU framework to realize the ensemble learning. The MPEL module can improve the reconstruction quality of HSI without additional training cost and inference time. We apply the MPEL module from the $4^{th}$ stage to $K^{th}$ stage.

Take the $(k+3)^{th}$ stage as an example, the MPEL module is described as follows. First, we use three $1 \times 1$ convolution functions to extract the image features from the denoising priors $\{v^k, v^{k+1}, v^{k+2}\}$ in the last three previous stages, then obtain $\{f_{e+3}^k, f_{e+3}^{k+1}, f_{e+3}^{k+2}\}$. After that, we fuse $\{f_{e+3}^k, f_{e+3}^{k+1}, f_{e+3}^{k+2}\}$ together, and the output of MPEL module is defined as

$$v_{mpe}^{k+2} = \gamma_{k+3} f_{e+3}^k + \gamma_{k+3} f_{e+3}^{k+1} + \gamma_{k+3} f_{e+3}^{k+2},$$

where $\{\gamma_{k+3}, \gamma_{k+3}, \gamma_{k+3}\}$ are trainable parameters that are initialized as $1/3$. These three parameters indicate the contributions of different denoising priors. Finally, the output of MPEL module is used as the input of projection-update step.

### IV. DEEU-NET FOR HSI RECONSTRUCTION

This section introduces the network structure of the 2D CNN-based denoiser, and then integrates the denoiser in the DEEU framework to form the DEEU-Net. In addition, the training method of the DEEU-Net is provided.

#### A. The structure of denoising network

According to Eq. (9), the DU method use a trainable deep neural network in the denoising step, which has a significant impact on the reconstruction performance [23]–[25]. It is well known that CNNs have been widely used in many computer vision tasks, including the 3D reconstruction [31] video restoration [16] image super-resolution [17] and object detection [18]. Moreover, 2D CNN has advantage in computational efficiency [19]. Thus, we plug a 2D CNN-based U-Net as denoiser in each stage to effectively improve the reconstruction quality of the proposed DEEU-Net, while retaining fast computation speed.

![Diagram of CNN-based denoiser](Image)

**Fig. 5.** The diagram of CNN-based denoiser used in this paper.

As shown in Fig. 5, this denoiser includes two down-sampling steps and two up-sampling steps. The number of feature channels is initialized as $32$, and the channel number is doubled after each down-sampling. Then, this denoiser is plugged in each stage of the DEEU framework, which forms the DEEU-Net to reconstruct the HSI cube from a single compressive measurement. According the Eq. (9), in the $k^{th}$ stage of DEEU-Net, the input of denoiser is the output of DEE module $f_{e_k}$, and the output of denoiser is the reconstructed HSI cube in this stage. This CNN-based denoiser is also applied in GAP-Net [24] and ADMM-Net [25], and we use this denoiser to assess the superiority of our proposed DEEU framework over other DU framework.

In addition, similar to the existing DU methods, our DEEU framework can work with more advanced denoising networks, which can obtain higher reconstruction quality [23]. However,
this may cost more inference time and limit the DU methods to achieve the real-time reconstruction.

Different from traditional DU methods, the trainable parameters of the proposed DEEU-Net include not only the network parameters of denoisers, but also the network parameters in the DEE modules and MPEL modules. Moreover, the network parameters at different stages are not shared.

B. Training method of DEEU-Net

The ultimate goal of DEEU-Net is to minimize the difference between the final output and the ground-truth HSI cube. Thus, the loss function for the network training can be defined as

\[ L_1 = \text{RMSE}(f, v^K), \]  

(21)

where \( \text{RMSE}(\cdot) \) represents the root mean square error (RMSE) and \( f \) is the ground-truth data.

Moreover, as mentioned in Section 3.2, our MPEL module not only improves the reconstruction performance, but also speeds up the convergence of DU models. However, if DU methods only uses a single denoiser prior in each stage, these models will become more difficult to train as the stage number increases. Therefore, in the following experiments, when the DU models do not use MPEL module, we will pre-train these models with another loss function as following

\[ L_2 = \eta \sum_k \text{RMSE}(f, v^{3\times k}) + L_1, \]  

(22)

where \( \eta \) is set to be 1/4 in this paper, and \( L_2(\cdot) \) calculates the loss value between the denoising prior and the ground-truth every three stages. Concretely, for the first half of the training process, we pre-train these models with \( L_2(\cdot) \). Then, we continue to fine-tune these models with \( L_1(\cdot) \).

V. EXPERIMENTS

This section compares the proposed DEEU-Net with several SOTA methods for CASSI reconstruction on both simulation and real experiments. In addition, several ablation experiments are performed to demonstrate the effectiveness of DEEU-Net and the contributions of different modules.

A. Experimental settings

1) Datasets: In the simulation experiments, two public hyperspectral datasets are used, namely CAVE [32] and KAIST [33]. The CAVE dataset includes 32 HSIs with spatial resolution \( 512 \times 512 \). Each HSI cube consists of 31 spectral channels, whose central wavelengths are ranging from 400nm to 700nm with 10nm interval. The KAIST dataset includes 30 HSIs with spatial resolution \( 3376 \times 2704 \). Each HSI cube consists of 31 spectral channels, whose central wavelengths are ranging from 420nm-720nm with 10nm interval. Following the same setting as Refs. [22], [28] and [29], we use CAVE as the training dataset, and select ten scenes from KAIST as the testing dataset.

In the real experiments, we adopt the same training dataset as the TSA-Net in Ref. [29]. Meanwhile, the testing dataset consisting of five scenes, which are real-captured measurements provided in Ref. [29]. The real CASSI system can capture 28 spectral channels ranging from 450nm to 650nm with 10nm interval, and the span distance \( ds = 54 \) (based on the system structure) [29]. Thus, the size of each measurement \( 660 \times 714 \), and the size of the corresponding HSI cube is \( 660 \times 660 \times 28 \).

2) Implementation details: To keep the simulation experiments consistent with the real experiments, we first interpolate the HSI cubes of the two public datasets into 28 spectral channels, ranging from 450nm to 650nm, and the shifting step \( \alpha \) of the dispersive prism is set as 2 pixels [20], [22], [28], [29]. In this paper, the sizes of the reconstruction HSIs in the simulation experiments and real experiments are \( 256 \times 256 \times 28 \) and \( 660 \times 660 \times 28 \), respectively. Thus, according to Eq. (4), the spatial sizes of the compressive measurements in the simulation experiments and real experiments are \( 256 \times 310 \) and \( 660 \times 714 \), respectively.

As shown in Figs. 2 and 3, a single compressive measurement, the system matrix and some other system parameters are regarded as inputs of the DEEU-Net. Equation (21) is used to calculate the loss value between the reconstructed HSI cube and the ground-truth. Compared to other SOTA deep learning methods, our model requires fewer epochs during the training process. That is, our DEEU-Net is trained with the Adam optimizer (\( \text{beta}_1 = 0.9 \) and \( \text{beta}_2 = 0.99 \)) for 200 epochs. The learning rate is initially set to be 0.001, which decreases by 50% in every 50 epochs.

3) Baselines: We compare the proposed DEEU-Net with other eleven SOTA methods for HSI reconstruction, including model-based methods and learning-based methods. The model-based methods include ADMM [34] and GAP-TV [13]. The learning-based methods include five end-to-end methods (A-Net [21], TSA-Net [29], HDNet [28], MST-L [22] and CST-L [20]) and four DU methods (DGSMMP [35], GAP-Net [24], ADMM-Net [25] and GAP-CCoT [23]). These comparative methods are achieved using the settings in their original papers.

Among the aforementioned methods, MST-L, CST-L and GAP-CCoT are the recent SOTA transformer-based methods that outperform other existing methods. Among the existing DU methods, ADMM-Net, GAP-Net and GAP-CCoT use the traditional linear projection methods, which are derived from the ADMM and GAP methods, respectively. Moreover, GAP-Net and ADMM-Net use the CNN-based denoiser in each stage, which was described in Section 4.1; GAP-CCoT uses the transformer-based denoiser in each stage [23]. Finally, the peak signal-to-noise ratio (PSNR) and structure similarity index measure (SSIM) are used as the metrics to evaluate the reconstruction performance of each method.

In addition, in order to fairly evaluate the advantages of our methods compared to other learning-based methods, all models adopt the same training setting in this paper. That is, during the training process of each model, data augmentation methods are used including the random rotation and flipping. Moreover, the number of training samples per epoch for each model is set to be 2000, and all models are implemented by the PyTorch framework and performed on an NVIDIA RTX
TABLE I
THE PSNR IN dB (UPPER ENTRY IN EACH CELL) AND SSIM (LOWER ENTRY IN EACH CELL) OF THE RECONSTRUCTION RESULTS OBTAINED BY THE PROPOSED DEEU-Net AND SEVERAL SOTA METHODS USING TEN TARGET SCENES. THE BEST RESULTS ARE MARKED IN BOLD FONT.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scene1</th>
<th>Scene2</th>
<th>Scene3</th>
<th>Scene4</th>
<th>Scene5</th>
<th>Scene6</th>
<th>Scene7</th>
<th>Scene8</th>
<th>Scene9</th>
<th>Scene10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>34.86</td>
<td>0.933</td>
<td>35.08</td>
<td>0.940</td>
<td>35.40</td>
<td>0.942</td>
<td>34.32</td>
<td>0.960</td>
<td>33.97</td>
<td>0.935</td>
<td>33.88</td>
</tr>
<tr>
<td></td>
<td>33.75</td>
<td>0.914</td>
<td>33.43</td>
<td>0.903</td>
<td>34.58</td>
<td>0.933</td>
<td>31.50</td>
<td>0.923</td>
<td>32.08</td>
<td>0.903</td>
<td>33.48</td>
</tr>
<tr>
<td></td>
<td>33.73</td>
<td>0.915</td>
<td>33.29</td>
<td>0.903</td>
<td>34.88</td>
<td>0.936</td>
<td>31.16</td>
<td>0.922</td>
<td>32.17</td>
<td>0.900</td>
<td>33.82</td>
</tr>
<tr>
<td></td>
<td>35.36</td>
<td>0.942</td>
<td>35.95</td>
<td>0.943</td>
<td>36.72</td>
<td>0.956</td>
<td>32.81</td>
<td>0.949</td>
<td>34.05</td>
<td>0.933</td>
<td>36.05</td>
</tr>
<tr>
<td></td>
<td>35.55</td>
<td>0.9420</td>
<td>36.05</td>
<td>0.950</td>
<td>37.33</td>
<td>0.970</td>
<td>33.17</td>
<td>0.957</td>
<td>34.35</td>
<td>0.9310</td>
<td>36.46</td>
</tr>
<tr>
<td></td>
<td>36.31</td>
<td>0.952</td>
<td>37.78</td>
<td>0.959</td>
<td>38.19</td>
<td>0.965</td>
<td>34.66</td>
<td>0.980</td>
<td>35.96</td>
<td>0.966</td>
<td>37.19</td>
</tr>
</tbody>
</table>

3090 GPU. The PSNR and SSIM results of each model are calculated by MATLAB.

B. Results based on simulation dataset
First, we evaluate different methods based on the simulation dataset. Table I shows the reconstruction PSNR and SSIM of the methods using ten target scenes. It is observed that the DEEU-Net yields impressive reconstruction performance. The DEEU-Net-6stage and DEEU-Net-9stage achieve 35.72 dB/0.950 (PSNR/SSIM) and 36.61 dB/0.960 (PSNR/SSIM), respectively.

Compared with the recent SOTA transformer-based methods, the average PSNR of DEEU-Net-6stage is 0.43 dB and 0.23 dB higher than MST-L and CST-L, respectively; the average PSNR of DEEU-Net-9stage is 1.32 dB and 1.12 dB higher, respectively. In addition, compared with the existing DU methods using the same CNN-based denoiser and more advanced transformer-based denoiser, our method also shows significant improvements. Specifically, DEEU-Net-9stage outperforms ADMM-Net-9stage, GAP-Net-9stage and the recent SOTA transformer-based GAP-CCoT-9stage by 2.32 dB, 3.20 dB and 1.43 dB, respectively. It is worth noting that the proposed method with fewer stages still achieves higher reconstruction performance than the other methods. That is, DEEU-Net-6stage surpasses ADMM-Net-9stage, GAP-Net-9stage and GAP-CCoT-9stage by 2.34 dB, 2.31 dB and 0.54 dB, respectively. These significant improvements demonstrate the effectiveness of the proposed DEEU-Net.

Figure 6 illustrates the reconstruction images of “Scene 2” obtained by DEEU-Net-9stage and several SOTA methods, where 4 out of 28 spectral channels are shown. Figures 6(a) and 6(b) show the RGB image and the compressive measurement, respectively. Figure 6(d) shows the enlarged images of the red boxes in the reconstructed HSIs (Fig. 6(e)). DEEU-Net-9stage gets better visual quality in the reconstructed images with more edge information and fewer artifacts, while preserving smoothness in spatial domain. In contrast, other methods either induce image blurring to lose texture information, or have undesired artifacts and blotches. In addition, Fig. 6(c) illustrates the spectral density curves of different reconstruction results, where the measurement location is located within the green box in the RGB image. Moreover, the spectral correlation values between the reconstructed spectral density curves and the ground-truth curve are provided. It is shown that DEEU-Net-9stage results in the highest spectral correlation, and thus the best spectral accuracy. These results demonstrate the advantages of the proposed DEEU-Net in both visual quality and spectral dimension-consistency reconstruction [20].
C. Ablation study

This section conducts a set of ablation experiments to validate the effectiveness of each component in the proposed DEEU-Net. First, we study the impact of the stage numbers in ADMM-Net, GAP-Net, GAP-CCoT and our DEEU-Net based on ten target scenes. In addition, we further analyze the contributions of DEE module and MPEL module in the proposed DEEU-Net. Table 2 shows the reconstruction performance metrics and the runtimes per sample for the aforementioned DU methods with 6 stages and 9 stages.

It is observed from Table 2, when the DEEU-Net-6stage only uses DEE module (but without using MPEL module), our DEEU-Net outperforms ADMM-Net, GAP-Net and GAP-CCoT by 2.15 dB, 2.31 dB and 0.48 dB, respectively. When the DEEU-Net-9stage only uses DEE module (but without using MPEL module), our DEEU-Net outperforms ADMM-Net, GAP-Net and GAP-CCoT by 2.18 dB and 2.15 dB and 0.38 dB, respectively. The superiority is mainly derived from the powerful ability of the proposed DEE module in learning the mapping between the compressive measurement and HSI cube. Specifically, different from the traditional linear projection methods, which are implemented by a few fixed matrix operations, the DEE module can leverage the powerful learning capabilities of deep learning techniques. Thus, when the HSI cube includes a large number of spectral channels, the DEE module shows better learning and generalization abilities.
TABLE II
AVERAGE RECONSTRUCTION PSNRs, SSIMs AND RUNTIMES OF DIFFERENT DEEP UNFOLDING METHODS OVER TEN TARGET SCENES.

<table>
<thead>
<tr>
<th>Method</th>
<th>ADMM-Net</th>
<th>GAP-Net</th>
<th>GAP-CCoT</th>
<th>DEEU-Net-6stage</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Running Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-stage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>32.86</td>
<td>0.914</td>
<td>0.0210</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>32.70</td>
<td>0.913</td>
<td>0.0187</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>34.53</td>
<td>0.940</td>
<td>0.0435</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td>35.01</td>
<td>0.948</td>
<td>0.0247</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>35.72</td>
<td>0.950</td>
<td>0.0253</td>
</tr>
<tr>
<td>9-stage</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td>33.38</td>
<td>0.921</td>
<td>0.0311</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td>33.41</td>
<td>0.922</td>
<td>0.0283</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td>35.06</td>
<td>0.949</td>
<td>0.0623</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td>35.56</td>
<td>0.952</td>
<td>0.0375</td>
</tr>
<tr>
<td></td>
<td>✓ ✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>36.61</td>
<td>0.960</td>
<td>0.0384</td>
</tr>
</tbody>
</table>

TABLE III
THE CONTRIBUTION OF EACH COMPONENT IN DEE MODULE FOR DEEU-Net-6stage.

<table>
<thead>
<tr>
<th>Component</th>
<th>Conv1 × 1</th>
<th>ConvBlock</th>
<th>SSFE block</th>
<th>Spatial FEB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPE block</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SSFE block</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td>34.44</td>
<td>34.66</td>
<td>35.23</td>
<td>35.72</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.940</td>
<td>0.944</td>
<td>0.950</td>
<td>0.950</td>
</tr>
</tbody>
</table>

In the following, we investigate the performance of different DU frameworks with simpler denoisers. The simpler denoiser has the same network structure as the denoiser in Fig. 5, but we halve the number of initial channels to 16. Then, the simpler denoiser is plugged into the ADMM, GAP and DEEU frameworks, using 6-stage architecture for each model. The average reconstruction PSNRs and SSIMs of these models are listed in Table 4. Compared with the results in Table 2, the average PSNRs obtained by ADMM-Net, GAP-Net and DEEU-Net decrease by 0.40 dB, 0.55 dB and 0.10 dB, respectively. It shows that the our DEEU-Net achieves the highest reconstruction quality and the least performance degradation. These results demonstrate the robustness of the proposed DEEU framework.

D. Results based on experimental dataset

This section assesses the reconstruction performance of DEEU-Net based on the real HSI dataset. The testing dataset are collected by the real CASSI system mentioned in Ref. [29]. Following the same settings as the existing methods [20], [23], [28], we re-train our DEEU-Net-9stage on the CAVE and KAIST datasets. Figure 7 shows the reconstruction results of our DEEU-Net-9stage and several representative SOTA methods with 4 (out of 28) spectral channels. Figures 7(a) and
Fig. 7. Reconstructed HSIs of “Scene 1” with 4 (out of 28) spectral channels using the real dataset. The proposed DEEU-Net-9stage is compared with several SOTA methods. (a) and (b) RGB image and compressive measurement. (c) shows the enlarged images of the white boxes in (d) reconstructed HSIs.

7(b) show the RGB image and the compressive measurement, respectively. Figure 7(d) shows the reconstructed HSIs. For each reconstructed image, we also magnify the region within the white box, which is shown in Fig. 7(c). Figure (c) corresponds to the white box in the RGB image. Compared to the other methods, DEEU-Net is able to recover more texture details and more complete background information with fewer image artifacts.

VI. CONCLUSION

This paper develops a novel learning-based DEEU framework for HSI reconstruction. The DEE module is used in the projection-update step to guide the iterative learning process. In the proposed DEE module, the efficient local estimation is implemented by the DPE block, and the SSFE block is then used to exploit the spectral correlation and spatial similarity of the HSI cube. DEE module shows better learning and generalization abilities than the traditional linear projection methods. In addition, the MPEL module is used to further improve the reconstruction quality without increasing runtime. Finally, the CNN-based denoiser is plugged in each stage to effectively improve the reconstruction quality of the proposed DEEU-Net, while retaining fast computation speed. The proposed methods are verified based on the experiments using both simulation and real datasets. It shows that our DEEU framework shows efficient reconstruction and generalization abilities than traditional DU methods employing linear projection. Moreover, our DEEU-Net is superior in both reconstruction quality and computation speed over the SOTA methods. More importantly, the proposed DEEU-Net explores the powerful deep learning technologies to guide the iterative learning process of the DU method for HSI reconstruction. In the future, we will design more efficient learning-based networks to further improve the HSI reconstruction performance.

REFERENCES


