

Deep plug-and-play self-supervised neural networks for spectral snapshot compressive imaging

ZHANG Xing-Yu^{1,3}, ZHU Shou-Zheng^{1,3}, ZHOU Tian-Shu^{1,3}, QI Hong-Xing^{1,3}, WANG Jian-Yu^{1,2,3},
LI Chun-Lai^{1,2,3*}, LIU Shi-Jie^{1,3*}

- (1. School of Physics and Optoelectronic Engineering, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, China;
2. Key Laboratory of Space Active Opto-Electronics Technology, Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China;
3. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: The encoding aperture snapshot spectral imaging system, based on the compressive sensing theory, can be regarded as an encoder, which can efficiently obtain compressed two-dimensional spectral data and then decode it into three-dimensional spectral data through deep neural networks. However, training the deep neural networks requires a large amount of clean data that is difficult to obtain. To address the problem of insufficient training data for deep neural networks, a self-supervised hyperspectral denoising neural network based on neighborhood sampling is proposed. This network is integrated into a deep plug-and-play framework to achieve self-supervised spectral reconstruction. The study also examines the impact of different noise degradation models on the final reconstruction quality. Experimental results demonstrate that the self-supervised learning method enhances the average peak signal-to-noise ratio by 1.18 dB and improves the structural similarity by 0.009 compared with the supervised learning method. Additionally, it achieves better visual reconstruction results.

Key words: compressed sensing, deep learning, self-supervised, coded aperture imaging

用于单次曝光压缩成像的深度即插即用自监督神经网络

张星宇^{1,3}, 朱首正^{1,3}, 周天舒^{1,3}, 齐洪兴^{1,3}, 王建宇^{1,2,3}, 李春来^{1,2,3*}, 刘世界^{1,3*}

- (1. 国科大杭州高等研究院 物理与光电工程学院, 浙江 杭州, 310024;
2. 中国科学院上海技术物理研究所 空间主动光电技术重点实验室, 上海 200083;
3. 中国科学院大学, 北京 100049)

摘要: 基于压缩感知理论的编码孔径快照式光谱成像系统可以看作编码器, 高效获取压缩后的二维光谱数据, 再通过深度神经网络解码为三维光谱数据。然而, 深度神经网络的训练需大量难以获得的干净数据。针对深度神经网络训练数据不足的问题, 提出一种基于邻域采样思想的自监督高光谱去噪神经网络, 并将其嵌入到深度即插即用框架中, 最终实现自监督光谱重建, 并验证不同噪声退化模型对最终重建质量的影响。实验表明, 在不需要干净数据作为标签的情况下, 自监督学习方法相较有监督学习方法的平均峰值信噪比提升 1.18 dB, 结构相似度提升 0.009, 且获得了更优的视觉重建效果。

关键词: 压缩感知; 深度学习; 自监督; 编码孔径成像

中图分类号: TP753

文献标识码: A

Received date: 2024-02-29, revised date: 2024-04-03

收稿日期: 2024-02-29, 修回日期: 2024-04-03

Foundation items: Supported by the Zhejiang Provincial "Jianbing" and "Lingyan" R&D Programs (2023C03012, 2024C01126)

Biography: Zhang Xing-Yu (1998-), male, Luohe, Henan Province, master. Research field is computational imaging. E-mail: zhangxingyu21@mails.ucas.ac.cn

*Corresponding authors: E-mail: liushijie@ucas.ac.cn; lichunlai@mail.sitp.ac.cn

Introduction

Spectral imaging technology enables the simultaneous acquisition of both spectral and image information^[1]. It stands as a crucial component of modern remote sensing, serving as a significant tool for both earth observation^[2] and space exploration^[3]. The three-dimensional spectral data cube contains a wealth of information, but its acquisition comes with higher costs, often necessitating multiple scans. In recent years, many advanced advances have been made in the field of computational imaging^[4]. The coded aperture snapshot spectral imager (CASSI) system offers a solution by compressing hyperspectral images using compressed sensing theory^[5]. This system modulates the spectral data cube, compresses it into two-dimensional measurements, and then utilizes algorithms for reconstruction. The physical structure of CASSI has two types, dual-disperser^[6] and single-disperser^[7]. Single-disperser structure has achieved applications in fields such as medicine^[8] due to its simpler physical design. The objective of compressed sensing image reconstruction is to restore the original high-dimensional signal $x \in R^N$ from a reduced set of linear measurements $y \in R^M$ ^[9]. Current compressed sensing reconstruction methods fall into two categories: model driven algorithms based on regularization prior and data-driven algorithms based on deep networks. Classical model driven algorithms include orthogonal matching pursuit^[10], iterative hard thresholding^[11-12] and block compressed sensing transforms^[13]. While model driven algorithms offer strong interpretability and generalization, it struggles with processing extensive spectral data, resulting in slower reconstruction speeds and poorer quality. In recent years, deep learning has evolved rapidly, with some technologies being applied to compressed sensing image reconstruction. Deep learning-based reconstruction algorithms have shown superior performance in both simulations and real-world scenarios^[14].

Xiong *et al.* pioneered the use of deep learning methods for hyperspectral compressive sensing reconstruction^[15]. They improved reconstruction using convolutional neural networks and residual connections. Choi *et al.* developed a convolutional autoencoder to obtain a nonlinear spectral representation of hyperspectral images, combining the learned autoencoder prior and total variation (TV) prior as a composite regular term, and solving the problem using the alternating direction multiplier method^[16]. Wang *et al.* designed HyperReconNet to reconstruct hyperspectral image by cascading spatial and spectral networks^[17]. Miao *et al.* proposed a λ -net for compressive sensing reconstruction of hyperspectral images and videos through two phases^[18]. They used Generative Adversarial Networks in the first phase of reconstruction and U-net in the second phase to enhance reconstruction. Meng *et al.* proposed TSA-net, incorporating a spatial-spectral self-attention module into U-net and integrating scatter-grain noise in the training process, significantly improving reconstruction of real data^[19]. Zheng *et al.* suggested a flexible plug-and-play (PnP)

framework for hyperspectral image reconstruction, enhancing both quality and speed^[20]. Wang *et al.* pioneered the application of the Transformer architecture to compressive sensing spectral imaging with the GAP-CSCoT network, maintaining high reconstruction quality while reducing runtime^[21]. Chen *et al.* proposed a Proximal Gradient Descent Unfolding Dense-spatial Spectral-attention Transformer (PGDUF) method that can accelerate the training of models based on the Transformer architecture without affecting the reconstruction results^[22]. Chen *et al.* utilised low-rank subspace representations of hyperspectral images in combination with deep neural networks to achieve better reconstruction results and stronger interpretability^[23]. Luo *et al.* proposed a Transformer-based HSI reconstruction method called dual-window multiscale Transformer (DWMT), which is a coarse-to-fine process, reconstructing the global properties of HSI with the long-range dependencies, and maintaining better reconstruction quality^[24].

By applying deep learning to compressive sensing reconstruction, high quality and fast reconstruction can be achieved through the powerful deep feature representation capability. However, most existing research on deep learning-based reconstruction algorithms is limited to supervised learning approaches, restricting their applicability in real-world scenarios. This is because supervised learning approaches require a large number of clean images to be used as labels, and acquiring clean images in the hyperspectral domain is expensive. In the realm of deep learning denoising, self-supervised learning has shown promise. In cases where the noise is zero-mean, the Noise2Noise method demonstrates that for a clean scene x and two independently noise-containing images y and z observed, a denoising network trained with (y, z) pairings is equivalent to a network trained with (y, x) pairings. Neighbor2Neighbor extends Noise2Noise by downsampling a single noise-containing image and training the two noise-containing sub-images obtained from downsampling as model inputs and labels. However, these self-supervised learning methods have only been tested for their denoising efficacy on RGB images or greyscale images and have not explored their performance on hyperspectral images.

In this paper, we propose a self-supervised compressive sensing hyperspectral image reconstruction algorithm based on PnP method and Neighbor2Neighbor strategy. Initially, we train a self-supervised hyperspectral denoising network using Neighbor2Neighbor, incorporating a channel attention mechanism to capture inter-spectral correlations in hyperspectral images. Subsequently, the denoising network is embedded into a deep plug-and-play framework based on the alternating multiplier method to achieve compressive sensing image reconstruction with a denoising model. We evaluate the algorithm's effectiveness in terms of both data metrics and visual effects, and compare the effects of different noise degradation models on the final reconstruction results.

The main contributions of this paper are as follows:

- (1) Based on Neighbor2Neighbor and SENet, we

propose a self-supervised hyperspectral image denoising network model Self-HSIDEcNN for subsequent compressed sensing image reconstruction.

(2) On the basis of Self-HSIDEcNN and PnP method, we propose PnP-Self-HSIDEcNN method to implement self-supervised hyperspectral compressed sensing image reconstruction.

(3) Through ablation experiments, we verified the effects of multiple hyperparameters in the PnP-Self-HSIDEcNN method on the reconstruction speed and reconstruction results, laying a foundation for its practical application.

The rest of this paper is organized as follows: section 1 describes the mathematical model of CASSI; section 2 introduces the mathematical principles of PnP method and Neighbor2Neighbor method, and introduces self-supervised denoising network Self-HSIDEcNN; section 3 demonstrates the effectiveness of our proposed self-HSIDEcNN method through extensive experiments; section 4 concludes the paper.

1 Mathematically model of CASSI

Let $X \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ be the spectral data cube, where x and y are the spatial dimensions and λ is the spectral dimension. Let $M^* \in \mathbb{R}^{N_x \times N_y}$ be the CASSI system physical mask, which can be regarded as a matrix of size $N_x \times N_y$. Each element of this matrix obeys a 0-1 distribution with probability p and is used to modulate the 3D spectral data cube signal. Let $X' \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ be the spectral data after passing through the mask plate, and for the n_λ -th band, we have

$$X'(:, :, n_\lambda) = X(:, :, n_\lambda) \odot M^* \quad (1)$$

where \odot represents the element-wise multiplication. After passing through the dispersion prism, the data cube X' is shifted in the y -axis. Let the offset signal be $X'' \in \mathbb{R}^{N_x \times (N_y + N_\lambda - 1) \times N_\lambda}$, and λ_c be the reference wavelength, we have

$$X''(u, v, n_\lambda) = X'(x, y + d(\lambda_n - \lambda_c), n_\lambda) \quad (2)$$

where (u, v) are the pixel coordinates in the plane of the detector, λ_n is the wavelength of the n_λ -th band, λ_c is the centre wavelength, $d(\lambda_n - \lambda_c)$ denotes the spatial offset of the n_λ -th band. This gives the measured value $y(u, v)$ at the position of the detector plane (u, v) is

$$y(u, v) = \int_{\lambda_{\min}}^{\lambda_{\max}} x''(u, v, n_\lambda) d\lambda \quad (3)$$

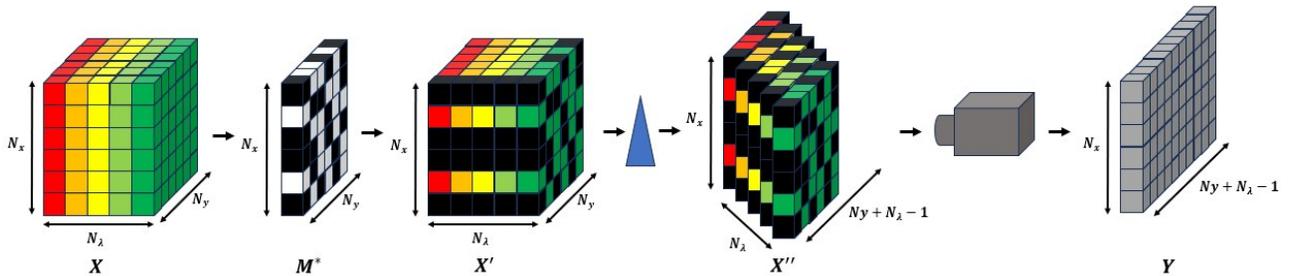


Fig. 1 CASSI forward model
图1 CASSI系统前向模型

The detector receives signals from all bands and finally obtains a two-dimensional measurement $Y \in \mathbb{R}^{N_x \times (N_y + N_\lambda - 1)}$. Considering the noise during the measurement, we have

$$Y = \sum_{n_\lambda=1}^{N_\lambda} X''(:, :, n_\lambda) + G \quad (4)$$

Let the offset physical mask plate matrix (this matrix can be fixed or variable^[25]) be $M \in \mathbb{R}^{N_x \times (N_y + N_\lambda - 1) \times N_\lambda}$, and the offset data cube is $\tilde{F} \in \mathbb{R}^{N_x \times (N_y + N_\lambda - 1) \times N_\lambda}$, we have

$$M(u, v, n_\lambda) = M^*(x, y + d(\lambda_n - \lambda_c)) \quad (5)$$

$$\tilde{F}(u, v, n_\lambda) = F(x, y + d(\lambda_n - \lambda_c), n_\lambda) \quad (6)$$

The final obtained measurement Y can be expressed as^[19]

$$Y = \sum_{n_\lambda=1}^{N_\lambda} \tilde{F}(:, :, n_\lambda) \odot M(:, :, n_\lambda) + G \quad (7)$$

The complete process is shown in Figure 1. The data obtained by CASSI is a two-dimensional measurement similar to Y . This data is characterised by a large amount of information and a small storage capacity. By means of appropriate algorithms, we were able to recover it as the original three-dimensional data cube.

2 Method

2.1 PnP method

The PnP method can decompose the original problem, which is difficult to solve, into subproblems that are easy to solve with good results^[26]. Hyperspectral compressed sensing image reconstruction can be conceptualized as the task of addressing the subsequent optimization problem:

$$\arg \min_x \frac{1}{2} \| \mathbf{y} - \mathbf{A}\mathbf{x} \|_2^2 + \lambda R(\mathbf{x}) \quad (8)$$

where \mathbf{y} is the measured value, \mathbf{x} is the original signal and $\lambda R(\mathbf{x})$ is the regularity term. The basic idea of PnP method for inverse problems is to use a pretrained denoiser for the desired signal as a prior. The method decomposes the whole problem into easier subproblems and solves the subproblems alternately in an iterative manner. The denoising network can be used as a flexible plug-in (i. e. , it can be easily changed) in the process. Specifically, problem (8) can be decomposed into the following subproblems using the alternating multiplier method^[27]:

$$\mathbf{x}^{k+1} = \arg \min_x \frac{1}{2} \| \mathbf{A}\mathbf{x} - \mathbf{y} \|_2^2 + \frac{\rho}{2} (\mathbf{x} - (\mathbf{z}^k - \mathbf{u}^k))_2^2 \quad (9)$$

$$\mathbf{z}^{k+1} = \arg \min_z \lambda R(\mathbf{z}) + \frac{\rho}{2} (\mathbf{z} - (\mathbf{x}^{k+1} + \mathbf{u}^k))_2^2 \quad (10)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (\mathbf{x}^{k+1} - \mathbf{z}^{k+1}) \quad , \quad (11)$$

where \mathbf{z} is the auxiliary variable, \mathbf{u} is the multiplier, ρ is the penalty factor, and k is the number of iterations. Let the auxiliary function $\text{prox}_g(\mathbf{v}) = \arg \min_x g(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{v}\|_2^2$, then Eqs. (9) - (11) can be written in the following form:

$$\mathbf{x}^{k+1} = \text{prox}_{\frac{f}{\rho}}(\mathbf{z}^k - \mathbf{u}^k) \quad , \quad (12)$$

$$\mathbf{z}^{k+1} = \text{prox}_{\frac{AR}{\rho}}(\mathbf{x}^{k+1} + \mathbf{u}^k) \quad , \quad (13)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (\mathbf{x}^{k+1} - \mathbf{z}^{k+1}) \quad , \quad (14)$$

where $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$. Eq. (12) has a solution in closed form, and Eq. (13) can be viewed as a denoising prior. The final PnP-ADMM solution for the compressive sensing hyperspectral reconstruction can be written as^[28]

$$\mathbf{x}^{k+1} = (\mathbf{z}^k - \mathbf{u}^k) + \frac{\mathbf{A}^\top [\mathbf{y} - \mathbf{A}(\mathbf{z}^k - \mathbf{u}^k)]}{[\text{Diag}(\mathbf{A}\mathbf{A}^\top) + \rho]} \quad , \quad (15)$$

$$\mathbf{z}^{k+1} = \mathcal{D}_{\hat{\sigma}_i}(\mathbf{x}^{k+1} + \mathbf{u}^k) \quad , \quad (16)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (\mathbf{x}^{k+1} - \mathbf{z}^{k+1}) \quad , \quad (17)$$

where $\sigma^2 = \lambda/\rho$ is the estimated noise bias and $\mathcal{D}_{\hat{\sigma}_i}$ is the denoiser. It should be noted that the performance of the noise reducer $\mathcal{D}_{\hat{\sigma}_i}$ directly affects the final reconstruction results. The noise reducer $\mathcal{D}_{\hat{\sigma}_i}$ used here in this paper is a self-supervised hyperspectral denoising network for the purpose of final self-supervised image reconstruction. During the practical application, the initial inputs consist of 2D measurements acquired from the detector and the mask matrix. Following several iterations of the pre-trained denoising network, the final reconstructed data cube is obtained. This process is shown in Fig. 2.

2.2 Neighbor2Neighbor method

The most critical part of the PnP-ADMM solution for compressive sensing hyperspectral reconstruction is the part of Eq. (16), which can be approximated using a deep learning model. Specifically it can be solved using

models in deep learning image denoising, but these models are mostly trained by supervised learning. There are two primary unsupervised deep learning denoising methods, one relies on deep image prior denoising method, which doesn't require training but has longer reconstruction times. The other is a denoising method based on Noise2Noise concept, which still needs training but can significantly diminish the reconstruction time.

The core idea of Noise2Noise is that for an unobserved clean scene x and two observed independent noise-containing images y and z , a noise reduction network trained with (y, z) pairings is equivalent to a network trained with (y, x) pairings, provided the noise is obeying a zero mean^[29]. The optimisation objective of Noise2Noise is

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{y}, \mathbf{z}} (f_{\theta}(\mathbf{y}) - \mathbf{z})_2^2 \quad , \quad (18)$$

where $f(\cdot)$ is the noise reduction network. Noise2Noise requires at least 2 separate noise-containing images for each scene, which is difficult to satisfy in real scenes. To increase the practical value of Noise2Noise, the theory of Noise2Noise is extended. For a single noisy image, one of the possible ways to construct two similar but not identical images is downsampling^[30].

The Neighbor2Neighbor downsampling idea is shown in Fig. 3. For a grayscale image y of size $H \times W$, divide it into $\frac{H}{2} \times \frac{W}{2}$ blocks of pixels of size 2×2 . Then two separate pixels are randomly sampled from each pixel block, which are finally combined to form two sub-sampled image $g_1(y)$ and $g_2(y)$ of size $\frac{H}{2} \times \frac{W}{2}$. At this point Eq. (18) becomes

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{y}} (f_{\theta}(g_1(\mathbf{y})) - g_2(\mathbf{y}))_2^2 \quad . \quad (19)$$

Reference [30] demonstrated that using Eq. (19) directly as the optimisation objective would end up with denoising results that are too smooth. Therefore, we consider adding a penalty term to Eq. (19) to get the final

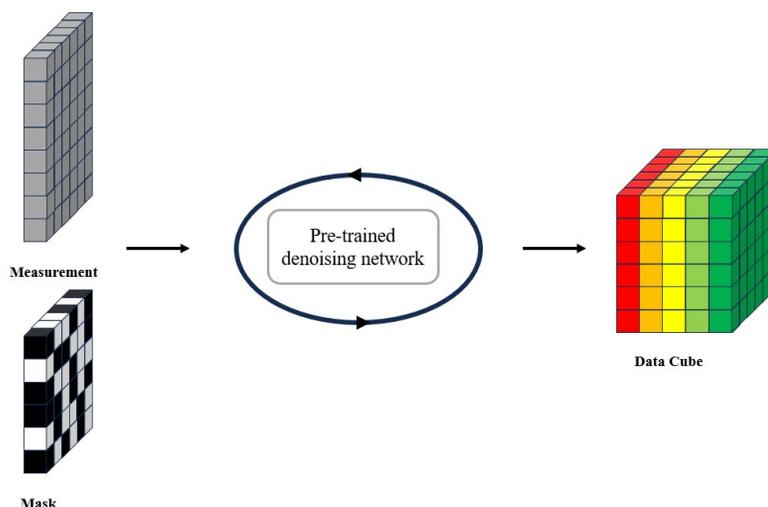


Fig. 2 PnP image reconstruction framework
图2 PnP 图像重建框架

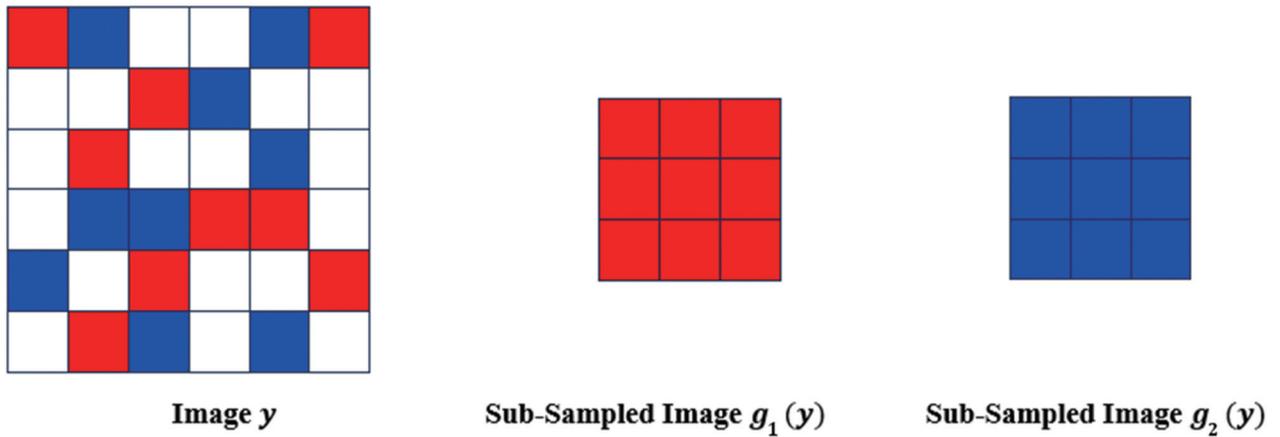


Fig. 3 Image downsampling method
图3 图像降采样方法

optimisation objective as

$$\begin{aligned} & \min_{\theta} \mathbb{E}_{x,y} \left(f_{\theta}(g_1(\mathbf{y})) - g_2(\mathbf{y}) \right)_2^2 \\ & + \gamma \mathbb{E}_{x,y} \left(f_{\theta}(g_1(\mathbf{y})) - g_2(\mathbf{y}) - g_1(f_{\theta}(\mathbf{y})) + g_2(f_{\theta}(\mathbf{y})) \right)_2^2 \end{aligned} \quad (20)$$

2.3 Hyperspectral denoising network

Deep denoising models for greyscale and RGB images have gained longevity in recent years. We propose deep plug-and-play self-supervised hyperspectral image denoising network (Self-HSIDeCNN). FFDNet is chosen as the base model framework for our model, which has the advantages of being flexible and fast, and has already shown excellent performance in denoising RGB images and greyscale images^[31]. We use FFDNet as a backbone network for two main reasons. Firstly, compared to UNet, a commonly used backbone network in image processing, FFDNet is designed for image denoising, and the noise level map can be varied during training to make the model flexible for various levels of noise. Second, compared with the transformer architecture, which has stronger feature extraction capability, FFDNet's training and inference are faster, and its downsampling module can effectively reduce the model's requirement on computational resources. The structure of our neural network is shown in Fig. 4.

In the process of performing frame-level noise reduc-

tion, for the n_{λ} -th band of the spectral data cube $\mathbf{X} \in \mathbb{R}^{N_x \times N_y \times N_{\lambda}}$, in addition to inputting this band of size $N_x \times N_y$ into the neural network, the images of its neighbouring K bands will also be inputted (in this paper, we take $K = 6$). For $K + 1$ band images at this point, following the sampling strategy in 2.2, two subgraphs, both of size $(K + 1) \times \frac{N_x}{2} \times \frac{N_y}{2}$, are obtained. One subgraph y_1 is used as model input and one subgraph y_2 is used as labels to compute the loss function. For subgraph y_1 , in order to speed up model training, another downsampling is performed after input to the model. Together with the noise estimation level map, the final model has an input size of $(4K + 5) \times \frac{N_x}{4} \times \frac{N_y}{4}$.

In order to better capture the correlation between different spectral bands of hyperspectral images, an inter-channel attention structure SENet is added to form a SEBlock after every two convolutional layers in the neural network. This structure is equivalent to assigning a separate weight to each channel feature map, which increases the number of parameters in the model, but enhances the non-local feature extraction capability of the model. The SENet structure is shown in Fig. 5. A total of seven SEBlocks are used in this paper, each using a convolutional kernel of size 3×3 and a ReLU activation function (the output of the last SEBlock does not use an activation func-

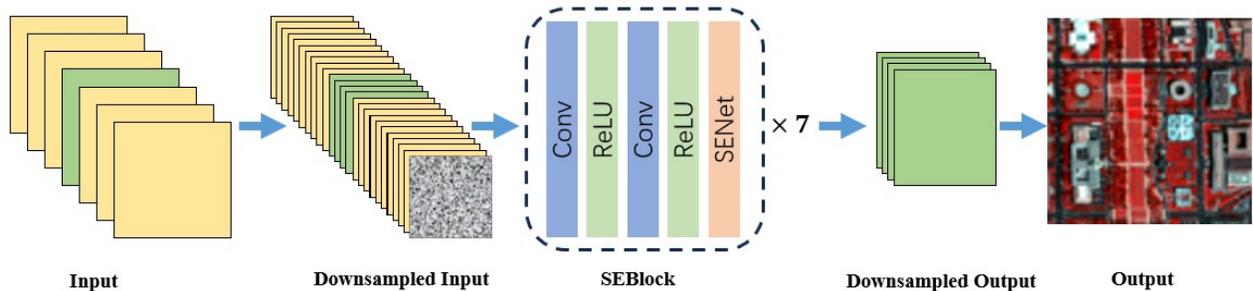


Fig. 4 Self-HSIDeCNN network architecture
图4 Self-HSIDeCNN 网络结构

tion). The feature map size is reduced to $(K + 1) \times \frac{N_x}{2} \times \frac{N_y}{2}$ by up-sampling after 7 SEBlocks, and then the loss function is calculated with the sub-sampled image y_2 . The noise level σ in the noise level estimation map is randomly changed during the training process, allowing the model to flexibly adapt to different noises.

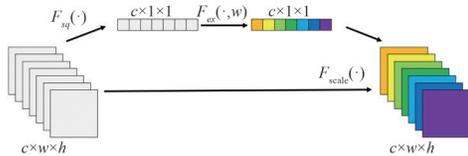


Fig. 5 SENet architecture
图5 SENet结构

After obtaining the hyperspectral denoising model, we can train the model based on the Neighbor2Neighbor self-supervised learning method. According to the optimization objective shown in Eq. (20), the loss function is designed as

$$L = \mathcal{L}_{\text{rec}} + \gamma \cdot \mathcal{L}_{\text{reg}}$$

$$= \left(f_{\theta}(g_1(\mathbf{y})) - g_2(\mathbf{y}) \right)_2^2$$

$$+ \gamma \cdot \left(f_{\theta}(g_1(\mathbf{y})) - g_2(\mathbf{y}) - \left(g_1(f_{\theta}(\mathbf{y})) - g_2(f_{\theta}(\mathbf{y})) \right) \right)_2^2, \quad (21)$$

where $g_{1,2}(\cdot)$ is a stochastic downsampling function and $f_{\theta}(\cdot)$ is a noise reduction neural network. γ is a penalty term (fixed at 5 in this paper) used to balance the level of detail preserved in the denoising results. To keep the gradient stable, the gradients of $g_1(f_{\theta}(\mathbf{y}))$ and $g_2(f_{\theta}(\mathbf{y}))$ are not propagated during training. This training process is shown in Fig. 6. In the inference stage, we only need to input the noisy image into the model to obtain the denoised image directly.

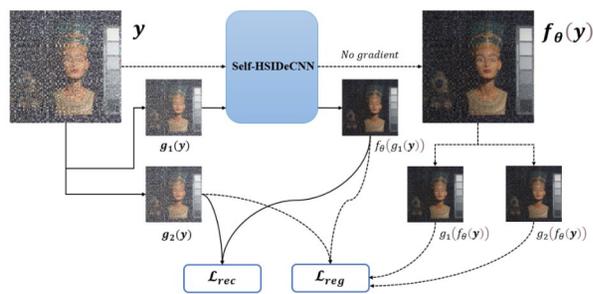


Fig. 6 Self-supervised training process
图6 自监督训练过程

For denoising models with supervised learning methods, the models cannot be trained in the absence of clean images as labels. For the compressed perceptual image reconstruction model with supervised learning method, the training data containing noise will make the model

worse (we will discuss this content in the subsequent section). As can be seen from Figure 6, this self-supervised learning method only needs noise-containing images to complete the training of the model, without the need of clean images as labels. The self-supervised learning approach is significantly superior in the hyperspectral domain where noise-free data is very expensive to obtain.

3 Experiments and results

Our proposed method can be divided into two steps. First, we train the hyperspectral denoising network based on the self-supervised approach. Then, this network is directly embedded into the PnP framework for self-supervised image reconstruction. In this section, we first describe the dataset and implementation details. Then, to evaluate the effectiveness, the proposed method is compared with models trained based on supervised learning approach. Furthermore, ablation studies are conducted to analyze the effect of hyperparameters on the results.

3.1 Training details

Model training was performed using the CAVE hyperspectral dataset, consisting of 32 scenes, each with a resolution of 512×512 and containing a total of 31 bands from 400 nm to 800 nm. Five of the scenarios were selected as the test set and the remaining scenarios as the training set. The RGB images of the five test scenes are shown in Fig. 7. To increase the training data for subsequent training, this dataset was randomly cropped and data augmented (including flipping, rotating, mirroring, and combinations of these operations). A total of 33,480 data of size $256 \times 256 \times 7$ were finally obtained for training. For the five test scenarios, the space was downsampled to a $256 \times 256 \times 31$ data cube. In this paper, the code was implemented using the PyTorch framework with 500 epochs using the Adam optimiser. The initial learning rate was set to 0.0001 and multiplied by 0.5 after every 100 epochs. Training of the entire network took approximately 8 hours, using a machine equipped with an Intel Xeon CPU, 360 GB of memory, and four Nvidia RTX 4090 Ti GPUs with 24 GB RAM.

3.2 Denoising results

Before comparing the reconstruction outcomes, it is essential to access both the supervised and the self-supervised denoising results to investigate the impact of the Neighbor2Neighbor self-supervised learning strategy on the denoising outcomes. The identical model, parameters, and training data are employed here, differing solely in the loss function and back-propagation gradient. For the supervised learning approach, the loss function is

$$L = \left(f_{\theta}(\mathbf{y}) - \mathbf{x} \right)_2^2. \quad (22)$$

For the self-supervised learning approach, the loss function is shown in Eq. (21).

With the maximum pixel value of 255, let the Gaussian noise variance be σ and the Poisson noise intensity be λ . In this paper, we compare four different noise degradation models, which are fixed Gaussian noise ($\sigma = 25$), range Gaussian noise ($\sigma \in [5, 50]$), fixed Poisson noise ($\lambda = 30$), and range Poisson noise ($\lambda \in [5, 50]$).



Fig. 7 RGB image of test scene
图7 测试场景RGB图像

Using peak signal to noise ratio (PSNR) and structural similarity (SSIM) as evaluation metrics. The five test scenario metrics were averaged and the final results are shown in Table 1.

Table 1 Comparison of denoising results
表1 去噪结果对比

Noise type	Supervised Model	Self-supervised Model
$\sigma = 25$	38.12 dB, 0.951	39.33 dB, 0.978
$\sigma \in [5, 50]$	37.63 dB, 0.924	39.16 dB, 0.969
$\lambda = 30$	38.28 dB, 0.964	40.25 dB, 0.983
$\lambda \in [5, 50]$	38.25 dB, 0.966	39.86 dB, 0.976

The self-supervised learning model, utilizing the Neighbor2Neighbor strategy, demonstrates superior performance compared to the supervised learning model across various noise degradation models. This superiority arises from the slight disparity ε between the labels of the self-supervised learning model and the desired denoising results of the model inputs. Additionally, with the incorporation of the penalty term in the loss function, the model achieves enhanced generalization and performs better on the test set.

The hyperparameter γ is used to avoid overly smooth denoising results. When $\gamma = 0$, it means that the penalty term in the loss function does not exist. In such a case, $f_{\theta}(g_1(\mathbf{y}))$ and $f_{\theta}(g_2(\mathbf{y}))$ are not exactly the same, and the model tends to output the average value of $f_{\theta}(g_1(\mathbf{y}))$ and $f_{\theta}(g_2(\mathbf{y}))$ because the loss function achieves its minimum value at this point. In order to verify the influence of the superparameter γ on the denoising results, the noise in the training data is unchanged (this paper uses the Gaussian noise with variance of 25 and the Poisson noise with intensity of 30), and only the value of the superparameter is changed to verify the denoising effect of different models on the test set. The final results are shown in Table 2.

For Gaussian noise, the denoising effect is first analysed when $\gamma = 0$ and $\gamma = 1$. When $\gamma = 0$, the penalty term does not exist, the denoising result will be too smooth, and the performance of Self-HSIDeCNN is not optimal. However, because of the presence of downsampling and upsampling modules in Self-HSIDeCNN, the denoising outcomes exhibit excessive smoothness in the downsampled subgraphs. However, despite this, the

Table 2 The influence of hyperparameter γ on denoising results

表2 超参数 γ 对去噪结果影响

γ	PSNR / dB	SSIM	PSNR / dB	SSIM
	($\sigma = 25$)	($\sigma = 25$)	($\lambda = 30$)	($\lambda = 30$)
0	39.27	0.970	40.26	0.983
1	39.23	0.962	40.23	0.981
2	39.26	0.965	40.34	0.986
5	39.33	0.978	40.25	0.983
20	39.30	0.977	39.89	0.979

high-frequency information in the final output image of the model remains well-preserved after upsampling the subgraphs. This explains why the denoising effect becomes worse at $\gamma = 1$ (the presence of the penalty term makes the denoising inadequate). At $\gamma = 5$, although a small amount of noise is not removed due to the increase in γ , the original information of the image is better recovered (compared to $\gamma = 0$). At $\gamma = 20$, more noise is retained along with the high-frequency detail information of the image, so the denoising effect becomes worse again.

For Poisson noise, the trend is similar to that of Gaussian noise, but its optimum is achieved at $\gamma = 2$. This is because the model has a better denoising effect on Poisson noise. At this point, only a smaller γ is needed to alleviate the problem of over-smoothing caused by the self-supervised learning method.

In this paper, we fix $\gamma = 5$ for experiments, and the value of γ should be determined according to the scene characteristics and experimental results in practical applications.

3.3 Reconstruction results

The self-supervised learning model Self-HSIDeCNN based on Neighbor2Neighbor in 3.2 can be directly embedded in deep plug-and-play architectures for compressive sensing image reconstruction. We name it PnP-Self-HSIDeCNN. Deep plug-and-play frameworks often require a warm start to speed up convergence. Here, the GAP-TV denoiser is used for 90 iterations first, and then the denoiser is switched to the self-supervised learning model for better reconstruction results. In addition, for the estimated noise level σ at the time of reconstruction, it was set to 30 regardless of the noise degradation model (normalised to 0 to 1). As the iteration progresses, this parameter can be gradually decreased to enhance the re-

construction quality. We employ another reconstruction algorithm named PnP-HSI, which utilizes a denoising model trained through supervised learning with the PnP framework.

To compare with end-to-end neural networks, we selected both U-net and TSA-net models. U-net consists of two main components: an encoder and a decoder. Each coding block contains two 3×3 convolutional layers and a 2×2 maximum pooling layer using the ReLU activation function. TSA-net, directly uses the structure from the Ref. [19] without further changes. For the dataset, the CAVE hyperspectral dataset is also used and its size is randomly cropped to 256×256 in the spatial dimension for data augmentation, resulting in 5000 training data of size $256 \times 256 \times 31$. The CASSI system forward model simulation was performed on the data cube before inputting the model, and the final model input was a 2D measurement of 256×316 . It is important to emphasize that our primary aim is to investigate the utilization of self-supervised learning in compressive sensing image reconstruction and to analyze the influence of various noise degradation models on the reconstruction outcomes. Consequently, we refrain from comparing our approach with current state-of-the-art models in terms of performance, as these models are trained using a supervised approach.

The final comparison results are shown in Table 3. In terms of reconstruction quality, the self-supervised reconstruction model proposed in this paper outperforms both U-net and TSA-net in terms of PSNR and SSIM of the reconstruction results. It can also be seen that the optimal results were achieved by the self-supervised learning model trained based on fixed Gaussian noise. This is because the mathematical form of \mathcal{D}_{σ_i} in Eq. (13) corresponds to a fixed Gaussian noise denoiser. Unless otherwise stated, the PnP-Self-HSIDEcNN used in the subsequent comparison experiments was trained based on Gaussian fixed noise ($\sigma = 25$).

Figure 8 depicts the visual representation of a test scenario. The image on the left exhibits the RGB image of the scene alongside the 2D measurements following simulation, while the image on the right showcases the reconstruction results obtained from various algorithms. In Fig. 9, an enlarged view of the last band image on the right side of Fig. 8 is presented to facilitate a visual comparison of its reconstruction quality.

In the evaluation of hyperspectral images, we not

only consider spatial metrics but also emphasize spectral metrics. Figure 10 illustrates the reconstruction of the spectral profile of the central position for this scene. For the spectral curves obtained from the reconstruction of different algorithms in Fig. 10(b), we use Spectral Angle Mapper (SAM) to evaluate their similarity with the real spectral curves, and this result is shown in the legend of Fig. 10(b). From Fig. 10, it can be seen that the reconstructed spectral curve based on PnP-Self-HSIDEcNN is most similar to the real spectral curve.

Based on the aforementioned metrics, it is evident that the self-supervised reconstruction algorithm proposed in this paper, based on the deep plug-and-play framework, yields superior results across various metrics. However, in terms of runtime, the end-to-end neural networks U-net and TSA-net hold a significant advantage, completing reconstruction in less than 1 second after training, whereas algorithms utilizing deep plug-and-play frameworks require several minutes due to the iterative process. Nevertheless, runtime durations at the minute level are generally acceptable in practical applications. Moreover, the method circumvents the necessity for clean data as labels during training, thereby substantially mitigating the issue of inadequate training data for compressive sensing deep learning reconstruction models in real-world scenarios.

3.4 Comparison of generalisability

In real-world application scenarios, data often contains noise, and ideally, clean data with minimal noise levels is preferred. In our proposed self-supervised reconstruction model, spectral data cubes containing noise can be obtained from real scenarios and then utilized for training to ensure generalization. However, for supervised deep learning models, training becomes challenging due to the absence of clean images as labels. One approach is to directly train with noisy images as labels, but this typically leads to a degradation in model performance. To evaluate the generalizability of the models, Gaussian noise was introduced to the CAVE dataset to simulate real-world scenarios. U-net and TSA-net were retrained and tested using this noisy data. Conversely, self-supervised networks, which inherently incorporate noise during the training process, can be directly tested with noisy data. The resulting performance is summarized in Table 4.

From Table 4, it's evident that the performance of the algorithms declines when tested on noisy data. How-

Table 3 Comparison of reconstruction results

表3 重建结果比较

	U-net	TSA-net	PnP-HSI	Ours ($\sigma = 25$)	Ours ($\sigma \in [5, 50]$)	Ours ($\lambda = 30$)	Ours ($\lambda \in [5, 50]$)
Scene1	26.29 dB, 0.843	26.47 dB, 0.855	29.56 dB, 0.875	31.12 dB, 0.892	30.44 dB, 0.843	28.34 dB, 0.788	28.51 dB, 0.798
Scene2	37.20 dB, 0.941	36.98 dB, 0.935	37.59 dB, 0.959	39.62 dB, 0.959	37.01 dB, 0.921	37.55 dB, 0.963	38.39 dB, 0.966
Scene3	33.87 dB, 0.918	35.07 dB, 0.917	36.27 dB, 0.924	35.89 dB, 0.916	34.31 dB, 0.888	34.14 dB, 0.908	34.25 dB, 0.909
Scene4	32.99 dB, 0.910	33.16 dB, 0.929	34.88 dB, 0.933	35.22 dB, 0.928	34.13 dB, 0.870	32.96 dB, 0.910	33.02 dB, 0.903
Scene5	20.11 dB, 0.788	21.14 dB, 0.794	21.55 dB, 0.797	23.89 dB, 0.838	23.94 dB, 0.808	23.06 dB, 0.810	22.70 dB, 0.802
Mean	30.09 dB, 0.880	30.56 dB, 0.886	31.97 dB, 0.898	33.15 dB, 0.907	31.97 dB, 0.866	31.21 dB, 0.876	31.37 dB, 0.876

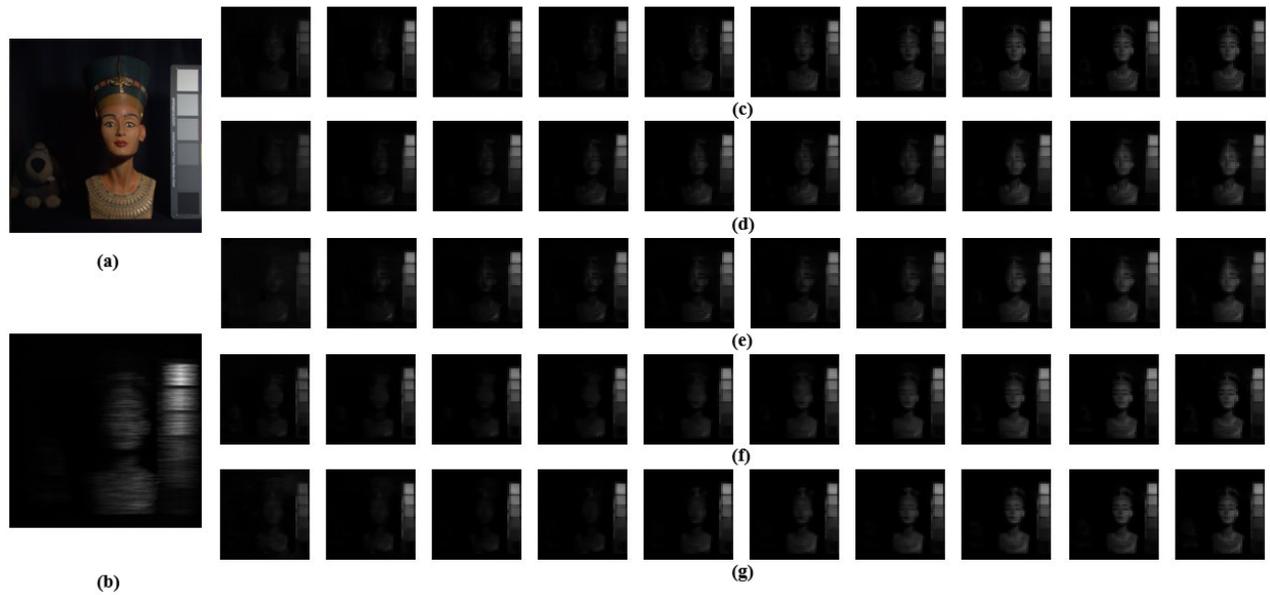


Fig. 8 Comparison of the visual effect of the reconstruction results of different algorithms: (a) RGB image; (b) 2D measurements; (c) ground truth; (d) U-net; (e) TSA-net; (f) PnP-HIS; (g) PnP-Self-HSIDeCNN
图8 不同算法重建结果视觉效果对比: (a)RGB 图像; (b)二维测量值; (c)真值图像; (d)U-net; (e)TSA-net; (f)PnP-HIS; (g) PnP-Self-HSIDeCNN

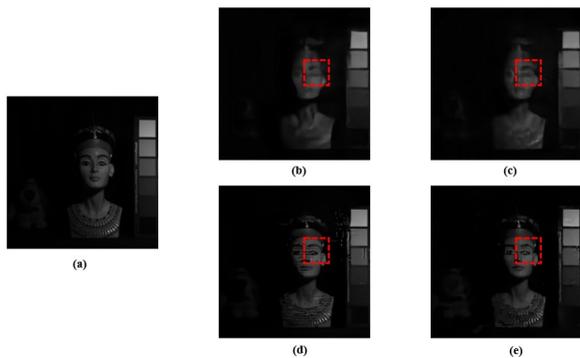


Fig. 9 Comparison of visual reconstruction effects in the final band: (a) ground truth; (b) U-net; (c) TSA-net; (d) PnP-HIS; (e) PnP-Self-HSIDeCNN
图9 最后一个谱段图像重建视觉效果比较: (a)真值图像; (b)U-net; (c)TSA-net; (d)PnP-HIS; (e)PnP-Self-HSIDeCNN

ever, the degradation in performance for the two algorithms based on the deep plug-and-play framework is considerably less than that observed for the two end-to-end neural networks. This indicates that the self-supervised learning model exhibits stronger robustness and generalization. Building upon the above findings, the self-supervised reconstruction algorithm proposed in this article, based on the deep plug-and-play framework, demonstrates superior performance across multiple indicators. Importantly, this method doesn't necessitate clean data as labels for training, thereby alleviating the challenge of insufficient training data for compressive sensing deep learning reconstruction models in practical applications.

3.5 Ablation experiment

3.5.1 Hyperparameterisation γ

In 3.2, we verified the effect of the hyperparameter

γ on the model's denoising results. Here, we explore its impact on the hyperspectral image reconstruction. Using the denoising model from Section 3.2 directly and maintaining other parameters in the PnP-Self-HSIDeCNN algorithm unchanged, we present the reconstruction results in Table 5.

For scenes 1 to 4, an appropriate penalty term can enhance the models' reconstruction quality. However, excessively large values of γ can degrade the model's denoising effect (even worse than without the penalty term), consequently affecting the final hyperspectral compressed perceptual image reconstruction. However, for scene 5, the model performance is better when γ is not zero. This is attributed to scene 5's rich colour information and more complex spatial-spectral curves, consequently affecting the final hyperspectral compressed perceptual image reconstruction. The comparison of Tables 2 and 5 reveals that the model's performance on the denoising task and on the image reconstruction task are not entirely correlated. On the denoising task, the model achieves optimal performance with $\gamma = 5$, but on the image reconstruction task the model achieves optimal performance with $\gamma = 2$. In summary, the value of γ should also be determined according to the specific usage scenario.

3.5.2 Noise estimation level σ

In the aforementioned comparison, the estimated noise level σ is set to 30, irrespective of the noise degradation model. This approach is chosen to facilitate direct model usage for reconstruction without the need for tedious parameter adjustment processes. Consequently, the reconstruction results of the PnP-Self-HSIDeCNN algorithm in the aforementioned experiments represent expected performance in practical use rather than optimal

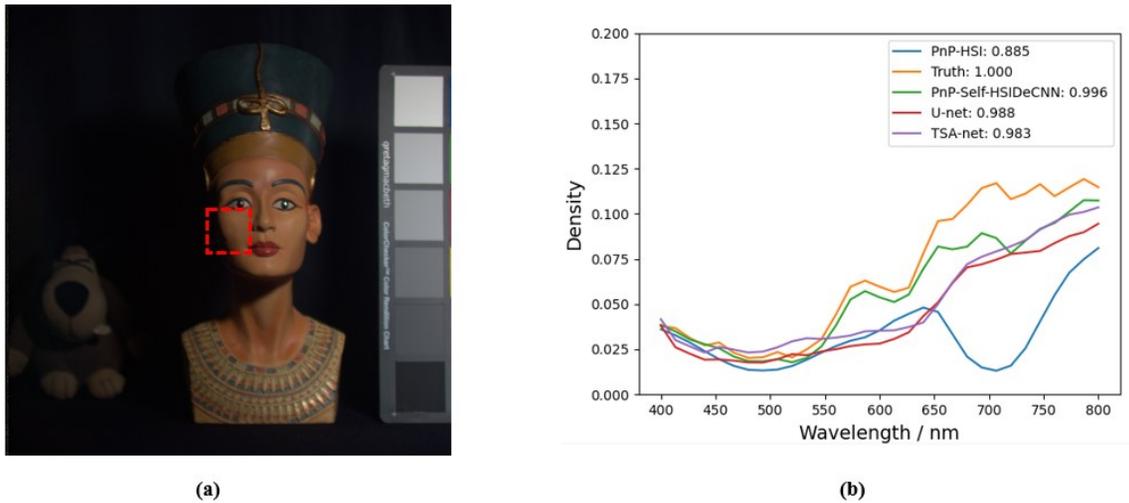


Fig. 10 Comparison of spectral curve reconstruction results; (a) RGB image; (b) spectral curve comparison
图10 光谱曲线重建效果对比: (a) RGB 图像; (b) 光谱曲线对比

Table 4 Comparison of generalisability
表4 泛化性比较

	U-net	TSA-net	PnP-HSI	Ours ($\sigma = 25$)
Scene1	25.89 dB, 0.824	25.88 dB, 0.838	29.52 dB, 0.873	31.13 dB, 0.890
Scene2	36.41 dB, 0.931	35.49 dB, 0.927	37.54 dB, 0.957	39.59 dB, 0.957
Scene3	31.78 dB, 0.887	33.80 dB, 0.904	36.23 dB, 0.919	35.84 dB, 0.912
Scene4	31.03 dB, 0.901	32.03 dB, 0.917	34.80 dB, 0.934	35.21 dB, 0.929
Scene5	20.14 dB, 0.773	20.50 dB, 0.787	21.56 dB, 0.798	23.84 dB, 0.834
Mean	29.05 dB, 0.863	29.54 dB, 0.874	31.93 dB, 0.899	33.12 dB, 0.904
Difference	-1.04 dB, -0.017	-1.02 dB, -0.012	-0.04 dB, -0.001	-0.03 dB, -0.003
Percentage	-3.46%, -1.93%	-3.34%, -1.35%	-0.13%, -1.11%	-0.09%, -0.33%

Table 5 Influence of hyperparameter γ on reconstruction results
表5 超参数 γ 对重建结果影响

	$\gamma = 0$		$\gamma = 1$		$\gamma = 2$		$\gamma = 5$		$\gamma = 20$	
	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM
Scene1	30.92	0.862	30.42	0.862	30.92	0.878	29.70	0.785	29.89	0.855
Scene2	37.19	0.921	38.54	0.940	40.19	0.951	35.50	0.921	37.67	0.944
Scene3	33.93	0.888	36.28	0.908	36.36	0.913	32.74	0.888	34.28	0.892
Scene4	33.98	0.883	35.07	0.899	35.47	0.894	33.19	0.861	35.38	0.916
Scene5	22.97	0.769	23.86	0.833	23.80	0.843	23.29	0.791	23.18	0.803
Mean	31.79	0.865	32.83	0.888	33.35	0.896	30.88	0.849	32.08	0.882

performance. To explore the model’s optimal performance, Gaussian fixed noise ($\sigma = 25$) is utilized during model training, and two noise estimation strategies are employed during model testing: fixed estimation of the noise level (fixed at 100, 50, 30, and 5, respectively), and dynamic estimation of the noise level (gradually decreasing as iterations progress). The experimental results are presented in Table 6 and Table 7.

When $\sigma = 100$, it significantly exceeds the noise level present in the training data for the denoising model. At this point, the results converge quickly but with poor

reconstruction quality. Smaller σ values yield better reconstruction results but require longer runtimes. Compared to fixed noise estimation levels, allowing the noise estimation level to gradually decrease with iterations achieves optimal reconstruction results, albeit at the expense of longer runtimes (since σ decreases as iterations progress). In summary, the value of σ should be selected based on specific requirements. Larger σ values can be used to expedite reconstruction when strict reconstruction quality is not necessary.

Table 6 Influence of noise estimation level σ on reconstruction results
表6 噪声评估水平 σ 对重建质量影响

	$\sigma = 100$		$\sigma = 50$		$\sigma = 30$		$\sigma = 5$		$\sigma \in [0, 50]$	
	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM	PSNR/dB	SSIM
Scene1	27.57	0.750	30.21	0.852	31.00	0.888	31.28	0.897	32.01	0.899
Scene2	33.56	0.921	38.16	0.941	39.38	0.955	40.18	0.973	40.32	0.977
Scene3	32.00	0.888	34.69	0.888	35.77	0.911	36.20	0.928	36.83	0.932
Scene4	31.87	0.861	34.59	0.912	35.13	0.927	35.50	0.929	36.14	0.941
Scene5	21.58	0.749	22.97	0.803	23.75	0.832	24.29	0.850	25.88	0.859
Mean	29.32	0.834	32.12	0.879	33.01	0.903	33.49	0.915	34.23	0.921

Table 7 Influence of noise estimation level σ on reconstruction time
表7 噪声评估水平 σ 对重建时间影响

	$\sigma = 100$	$\sigma = 50$	$\sigma = 30$	$\sigma = 5$	$\sigma \in [0, 50]$
Scene1	53.10 s	96.57 s	99.09 s	105.39 s	117.74 s
Scene2	51.21 s	91.74 s	95.67 s	101.17 s	110.92 s
Scene3	48.69 s	84.51 s	89.87 s	94.28 s	105.03 s
Scene4	51.84 s	91.35 s	96.27 s	103.77 s	116.19 s
Scene5	55.62 s	104.61 s	112.84 s	119.07 s	145.56 s
Mean	52.09 s	93.76 s	98.75 s	104.74 s	119.09 s

3.5.3 Number of warm start iterations

Deep plug-and-play frameworks typically require a hot-start to expedite convergence. In the aforementioned experiments, the GAP-TV denoiser is employed for 80 iterations before transitioning to a self-supervised learning model to achieve improved reconstruction results. However, the GAP-TV noise reducer doesn't necessarily need 80 iterations to reach its denoising performance limit. Thus, reducing the number of hot-start iterations could accelerate reconstruction. However, ending hot-start iterations prematurely may result in suboptimal final image reconstruction. To investigate this, we select a scene with a fixed noise estimation level of 30 and vary the number of hot-start iterations to study their effects on reconstruction results and speed. The experimental results are presented in Table 8.

Table 8 Influence of the number of warm start iterations
表8 热启动迭代次数对重建结果影响

Number of warm start iterations	Warm start time / s	Self-supervised reconstruction time / s	Total reconstruction time / s	PSNR / dB	SSIM
0	0	39.06	39.06	33.79	0.828
10	5.24	40.95	46.19	34.33	0.867
20	10.04	39.69	49.73	34.43	0.867
30	15.76	40.95	56.71	34.44	0.871
40	20.68	40.32	61.00	34.42	0.869
50	26.41	40.32	66.73	34.25	0.869
60	31.23	39.69	70.92	34.21	0.867
70	36.44	39.69	76.13	34.17	0.864
80	41.67	40.32	81.99	34.15	0.869
90	46.88	40.95	87.83	34.12	0.869
100	52.27	39.69	91.96	34.12	0.867

As can be seen from Table 8, regardless of the number of hot-start iterations, the running time of the image reconstruction phase of the PnP-Self-HSIDEcNN using the deep model is basically unchanged. Hence, the total running time of the model is primarily limited by the hot-start time. The total image reconstruction time is shortest when no hot-start is performed, but this compromises the final reconstruction results. As the number of hot-start iterations increases, the image reconstruction quality decreases slightly and the total reconstruction time increases. Therefore, in practice, the number of hot-start iterations can be set to 10-20 to accelerate reconstruction while ensuring satisfactory reconstruction quality.

4 Conclusions

In this paper, we propose a self-supervised learning method, PnP-Self-HSIDEcNN, based on a deep plug-and-play framework and neighborhood sampling strategy. Unlike traditional methods, it only requires noisy images for training and fully exploits the inter-spectral correlation of hyperspectral images through the SENet structure. Comparing the results across visual evaluation, PSNR, SSIM, and previous end-to-end neural network reconstructions, the self-supervised learning method proposed herein achieves commendable reconstruction results with robust generalization within acceptable runtimes. We conduct a detailed and comprehensive comparison test on three hyperparameters of the loss func-

tion — penalty factor γ , noise level estimation parameter σ , and number of hot-start iterations — to fully explore their impacts on final reconstruction results in terms of speed and quality, thereby laying a relevant foundation for the algorithm's future practical applications. Our findings demonstrate that self-supervised learning can yield satisfactory performance even with limited or poor-quality data, providing a feasible approach for the future application of compressed sensing and CASSI systems in real-world scenes. However, compared to end-to-end neural networks, our approach lacks in real-time performance. Also, real-world noise is far more complex than Gaussian and Poisson noise. For future work, we would like to reduce the iteration time of our method and extend it to real-world data.

References

- [1] Wang Jian-Yu, Shu Rong, Liu Yin-nian, *et al.* Introduction to imaging spectroscopy [M]. Science Press (王建宇, 舒嵘, 刘银年, 等. 像光谱技术导论[M]. 科学出版社), 2011:1-3, 105-107.
- [2] Wu P Z. Characteristics and applications of satellite-borne hyperspectral imaging spectrometer [J]. *Remote Sensing of Land Resources*, 1999(3):10.
- [3] Ouyang Z Y. Ouyang Ziyuan: Scientific objectives of China's lunar exploration project [J]. *Proceedings of the Chinese Academy of Sciences* (欧阳自远. 中国探月工程的科学目标[J]. 中国科学院院刊), 2006, (05): 370-371.
- [4] Dun X, Fu Q, Li H T, *et al.* Advances in the frontiers of computational imaging [J]. *Chinese Journal of Image Graphics*, 2022, 27(6): 37.
- [5] Donoho D L. Compressed sensing [J]. *IEEE Transactions on Information Theory*, 2006, 52(4): 1289-1306.
- [6] Gehm M E, John R, Brady D J, *et al.* Single-shot compressive spectral imaging with a dual-disperser architecture [J]. *Optics Express*, 2007, 15(21): 14013-14027.
- [7] Wagadarikar A, John R, Willett R, *et al.* Single disperser design for coded aperture snapshot spectral imaging [J]. *Applied Optics*, 2008, 47(10):B44-51.
- [8] Meng Z, Qiao M, Ma J, *et al.* Snapshot hyperspectral endomicroscopy[J]. *Optics Letters*, 2020, 45(14).
- [9] Candes E J, Wakin M B. An introduction to compressive sampling [J]. *IEEE Signal Processing Magazine*, 2008, 25(2):21-30.
- [10] Tropp J A, Gilbert A C. Signal recovery from random measurements via orthogonal matching pursuit [J]. *IEEE Transactions on Information Theory*, 2007, 53(12):4655-4666.
- [11] Blumensath T, Davies M E. Iterative hard thresholding for compressed sensing [J]. *Applied & Computational Harmonic Analysis*, 2009, 27(3):265-274.
- [12] Carrillo R.E, Barner, *et al.* Lorentzian iterative hard thresholding: robust compressed sensing with prior information [J]. *Signal Processing*, 2013, 61(19): 4822-4833.
- [13] Mun S, Fowler J E. Block compressed sensing of images using directional transforms [C]. Image Processing (ICIP 2009), 2009.
- [14] Sun Y, Chen J, Liu Q, *et al.* Learning image compressed sensing with sub-pixel convolutional generative adversarial network [J]. *Pattern Recognition*, 2019, 98(12): 107051.
- [15] Xiong Z, Shi Z, Li H, *et al.* HSCNN: CNN-based hyperspectral image recovery from spectrally undersampled projections [J]. *IEEE Computer Society*, 2017:518-525.
- [16] Choi I, Jeon D S, Nam G, *et al.* High-quality hyperspectral reconstruction using a spectral prior [C]. International Conference on Computer Graphics and Interactive Techniques. ACM, 2017.
- [17] Wang L, Zhang T, Fu Y, *et al.* HyperReconNet: joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging [J]. *IEEE Transactions on Image Processing*, 2019, 28(5): 2257-2270.
- [18] Miao X, Yuan X, Pu Y, *et al.* l-net: Reconstruct hyperspectral images from a snapshot measurement [C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 4059-4069.
- [19] Meng Z, Ma J, Yuan X. End-to-end low cost compressive spectral imaging with spatial-spectral self-attention [C]. European Conference on Computer Vision. Springer, Cham, 2020.
- [20] Zheng S, Liu Y, Meng Z, *et al.* Deep plug-and-play priors for spectral snapshot compressive imaging [J]. *Photonics Research*, 2020, 9(2): I0011-I0022.
- [21] Wang L, Wu Z, Zhong Y, *et al.* Spectral compressive imaging reconstruction using convolution and spectral contextual transformer [J]. *arXiv e-prints*, 2022.
- [22] Chen Z, Cheng J. Proximal gradient descent unfolding dense-spatial spectral-attention transformer for compressive spectral imaging [J]. *arXiv preprint arXiv:2312.16237*, 2023.
- [23] Chen Y, Lai W, He W, *et al.* Hyperspectral compressive snapshot reconstruction via coupled low-rank subspace representation and self-supervised deep network [J]. *IEEE Transactions on Image Processing*, 2024.
- [24] Luo F, Chen X, Gong X, *et al.* Dual-window multiscale transformer for hyperspectral snapshot compressive imaging [C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(4): 3972-3980.
- [25] Takabe T, Han X H, Chen Y W. Deep versatile hyperspectral reconstruction model from a snapshot measurement with arbitrary masks [C]. ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 2390-2394.
- [26] Yang S, Ding X, Yuan H, *et al.* Reconstruction quality evaluation of compressed sensing image mapping spectrometer [C]. AOPC 2023: Computing Imaging Technology. SPIE, 2023, 12967: 57-65.
- [27] Boyd S, Parikh N, Chu E, *et al.* Distributed optimization and statistical learning via the alternating direction method of multipliers [J]. *Foundations & Trends in Machine Learning*, 2010, 3(1):1-122.
- [28] Yuan X. Generalized alternating projection based total variation minimization for compressive sensing [J]. *IEEE*, 2015.
- [29] Lehtinen J, Munkberg J, Hasselgren J, *et al.* Noise2Noise: learning image restoration without clean data [J]. *arXiv e-prints*: 1803.04189, 2018.
- [30] Huang T, Li S, Jia X, *et al.* Neighbor2neighbor: Self-supervised denoising from single noisy images [C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 14781-14790.
- [31] Zhang K, Zuo W, Zhang L. FFDNet: Toward a fast and flexible solution for cnn based image denoising [J]. *IEEE Transactions on Image Processing*, 2017: 1-1.