A Trainable Spectral-Spatial Sparse Coding Model for Hyperspectral Image Restoration

Anonymous Author(s)
Affiliation
Address
email

Abstract

Hyperspectral imaging offers new perspectives for diverse applications, ranging from the monitoring of the environment using airborne or satellite remote sensing, precision farming, food safety, planetary exploration, or astrophysics. Unfortunately, the spectral diversity of information comes at the expense of various sources of degradation, and the lack of accurate ground-truth "clean" hyperspectral signals acquired on the spot makes restoration tasks challenging. In particular, training deep neural networks for restoration is difficult, in contrast to traditional RGB imaging problems where deep models tend to shine. In this paper, we advocate instead for a hybrid approach based on sparse coding principles that retain the interpretability of classical techniques encoding domain knowledge with handcrafted image priors, while allowing to train model parameters end-to-end without massive amounts of data. We show on various denoising benchmarks that our method is computationally efficient and significantly outperforms the state of the art.

1 Introduction

Hyperspectral imaging (HSI) enables measurements of the electromagnetic spectrum of a scene on multiple bands (typically about a hundred or more), which offers many perspectives over traditional color RGB imaging. For instance, the high-dimensional information present in a single pixel is sometimes sufficient to identify the signature of a particular material, which is of course infeasible in the RGB domain. Not surprisingly, hyperspectral imaging is then of utmost importance and has a huge number of scientific and technological applications such as remote sensing [8, 21, 42], quality evaluation of food products [15, 18, 31], medical imaging [2, 17, 33], agriculture and forestry [1, 32, 36], microscopy imaging in biology [24, 52], or exoplanet detection in astronomy [23].

Information contained in hyperspectral signals is much richer than in RGB images, but the price to pay is the need to deal with complex degradations that may arise from multiple sources, including sparse noise with specific patterns (stripes), in addition to photon and thermal noise [27, 47]. As a consequence, HSI denoising is a crucial pre-processing step to enhance the image quality before using data in downstream tasks such as semantic segmentation or spectral unmixing [28]. A second issue is the lack of large-scale collection of ground-truth high-quality signals and the large diversity of sensor types, which makes it particularly challenging to train machine learning models for restoration such as convolutional neural networks. To deal with the scarcity of ground-truth data, most successful approaches today typically encode strong prior knowledge about data within the model architecture, which may be low-rank representations of input patches [16, 22, 49, 55, 64], sparse coding [12, 20, 22], or image self-similarities [35, 45, 65], which have proven to be very powerful in the RGB domain [9].

In this paper, we propose a fully interpretable machine learning model for hyperspectral images that may be seen as a hybrid approach between deep learning techniques, where parameters can be learned end to end with supervised data, and classical methods that essentially rely on image priors.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Since designing an appropriate image prior by hand is very hard, our goal is to benefit from deep learning principles (here, differentiable programming [6]) while encoding domain knowledge and physical rules about hyperspectral data directly into the model architecture, which we believe is a key to develop robust approaches that do not require massive amounts of training data.

More precisely, we introduce a novel trainable spectral-spatial sparse coding model with two layers, which performs the following operations: (i) The first layer decomposes the spectrum measured at each pixel as a sparse linear combination of a few elements from a learned dictionary, thus performing a form of linear spectral unmixing per pixel, where dictionary elements can be seen as basis elements for spectral responses of materials present in the scene. (ii) The second layer builds upon the output of the first one, which is represented as a two-dimensional feature map, and sparsely encodes patches on a dictionary in order to take into account spatial relationships between pixels within small receptive fields. To further reduce the number of parameters to learn and leverage classical prior knowledge about spectral signals [55], we also assume that the dictionary elements admit a low-rank structure—that is, dictionary elements are near separable in the space and spectrum domains, as detailed later. Even though dictionary learning has been originally introduced for unsupervised learning [38, 43], we adopt an unrolled optimization procedure inspired by the LISTA algorithm [25], which has been very successful in imaging problems for training sparse coding models from supervised data [30, 51, 58].

Our motivation for adopting a two-layer model is to provide a shared architecture for different HSI sensors, which often involve a different number of bands with different spectral responses. Our solution consists of learning sensor-specific dictionaries for the first layer, while the dictionary of second layer is shared across modalities. This allows training simultaneously on several HSI signals, the first layer mapping input data to a common space, before processing data by the second layer.

We experimentally evaluate our HSI model on standard denoising benchmarks, showing a significant improvement over the state of the art (including deep learning models and more traditional baselines), while being computationally very efficient at test time. Perhaps more important than pure quantitative results, we believe that our work also draws interesting conclusions for machine learning. First, by encoding prior knowledge within the model architecture directly, we obtain models achieving excellent results with a relatively small number of parameters to learn, a conclusion also shared by [30] for RGB imaging problems; nevertheless, the effect is stronger in our work due to the scarcity of training data for HSI denoising and the difficulty to train deep learning models for this task. Second, we also show that interpretable architectures are useful: our model architecture can adapt to different noise levels per band and modify the encoding function at test time in a principled manner, making it well suited for solving blind denoising problems that are crucial for processing hyperspectral signals.

2 Related Work on Hyperspectral Image Denoising

Learning-free approaches. Classical signal processing models for grayscale image denoising such as BM3D [11] may be applied indpendently to each spectral band of HSI signals. Not surprisingly, such an approach fails to capture relations between different channels and multi-band techniques have been shown to perform better. An extension of BM3D to volumetric data is proposed with BM4D [35], which is more relevant for HSI. GLF [65] builds upon the same principles as BM3D—namely, exploiting self-similarities by grouping 3D patches and collaborative filtering—while introducing subspace identification and data-adaptive filtering based on principal component analysis. GLF appears to be a very strong baseline, but is unfortunately very computationally demanding at test time. Finally, other approaches adapt traditional image processing priors such as total variation [59, 54], or wavelet sparsity [44, 48] to HSI denoising, but they tend to perform worse than GLF.

Sparse coding and low-rank models. Sparse coding and dictionary learning for natural image patches have been introduced in neuroscience by Olshausen and Field [43]. This is an unsupervised learning technique consisting of representing a signal as a linear combination of a few elements from a learned dictionary, which has shown to be very effective for various image restoration tasks [14, 39]. For hyperspectral images, another powerful and classical prior is the low-rank assumption about image patches, which is natural since spatial structures are often shared among multiple spectral bands [16, 49, 61]. Several approaches have then combined dictionary learning and low-rank regularization. For instance, 3D patches are represented as tensors in [45] and are encoded by using spatial-spectral dictionaries involving a Tucker decomposition [53]. In [64], 2D patches are extracted from the band-vectorized representation of the 3D HSI data and sparsely encoded on a

dictionary, while encouraging low-rank representations with a trace norm penalty on the reconstructed image. The low-rank constraint can also be enforced by designing the dictionary as the result of the matrix multiplication between spatial and spectral dictionaries learned by principal component analysis as in [20]. However, these methods typically compute sparse representations with an iterative optimization procedure, which may be computationally demanding at test time.

Deep learning. Like BM3D above, convolutional neural networks for grayscale image denoising (*e.g.*, DnCNN [62]) may also be applied to each spectral band, which is of course suboptimal. Because deep neural networks have been highly successful for RGB images with often low computational inference cost, there have been many attempts to design deep neural networks dedicated to HSI denoising. For instance, to account for the large number of hyperspectral bands, several approaches based on convolutional neural networks are operating on sliding windows in the spectral domain, [34, 50, 60], which allows training models on signals with different number of spectral bands, but the sliding windows significantly increases the inference time. More precisely, attention layers are used in [50], while more traditional CNNs are used in [34], possibly with residual connections [60]. Recently, an approach based on recurrent architecture was proposed in [57] to process signals with an arbitrary number of bands, achieving impressive results for various denoising tasks.

Hybrid approaches. SMDS-Net [58] adopts a hybrid approach between sparse coding and deep learning models by adapting the RGB image restoration method of [30] to HSI images. The resulting pipeline however lacks interpretability: SMDS-Net first denoises the input image with non-local means [9], then performs subspace projection in the spectral domain using HySime[7], before sparsely encoding 3D patches (cubes) with a trainable version of Tensor-based ISTA [46]. Although this method reduces considerably the number of parameters in comparison to vanilla deep learning models, the spectral sliding windows approach still lacks physical interpretability, since the same denoising procedure is applied across different bands, which may not suffer from the same level of noise. Furthermore, stacking multiple denoising algorithms together increases both the computational cost and the intrinsic complexity of the approach, as shown in our experiments. In contrast, we propose a much simpler sparse coding model, which is fully interpretable and physically consistent with the nature of hyperspectral signals, by introducing a novel differentiable low-rank sparse coding layer.

3 A Trainable Spectral-Spatial Sparse Coding Model

In this section, we introduce our trainable spectral-spatial sparse coding model dedicated to hyperspectral imaging, and start by presenting some preliminaries on sparse coding.

3.1 Background on Sparse Coding

Image denoising with dictionary learning. A classical approach introduced by Elad and Aharon [14] for image denoising consists in considering the set of small overlapping image patches (*e.g.*, 8×8 pixels) from a noisy image, and compute a sparse approximation of these patches onto a learned dictionary. The clean estimates for each patch are then recombined to produce the full image.

Formally, let us consider a noisy image $\mathbf{y}$ in $\mathbb{R}^{c \times h \times w}$ with c channels and two spatial dimensions. We denote by $\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_n$ the n overlapping patches from $\mathbf{y}$ of size $c \times s \times s$, which we represent as vectors in $\mathbb{R}^m$ with $m = cs^2$. Assuming that a dictionary $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_p]$ in $\mathbb{R}^{m \times p}$ is given—we will discuss later how to obtain a "good" dictionary— each patch $\mathbf{y}_i$ is processed by computing a sparse approximation:

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{1}$$

where $\| \cdot \|_1$ is the l_1-norm, which is known to induce sparsity in the problem solution [38], and $\boldsymbol{\alpha}_i$ is the sparse code representing the patch $\mathbf{y}_i$, while λ controls the amount of regularization. Note that the ℓ_0-penalty, which counts the number of non-zero elements, could also be used, leading to a combinatorial problem whose solution is typically approximated by a greedy algorithm. After solving the n problems (1), each patch $\mathbf{y}_i$ admits a "clean" estimate $\mathbf{D}\boldsymbol{\alpha}_i$. Because each pixel belongs to several patches, the full denoised image $\hat{\mathbf{x}}$ is obtained by averaging these estimates.

Finding a good dictionary can be achieved in various manners. In classical dictionary learning algorithms, $\mathbf{D}$ is optimized such that the sum of the loss functions (1) is as small as possible, see [38] for a review. Adapting the dictionary with supervision is also possible [37], as discussed next.

Differentiable programming for sparse coding. The proximal gradient descent method called ISTA [19] is a classical algorithm for solving the Lasso problem in Eq. (1), which consists of the following iterations

$$\boldsymbol{\alpha}_i^{(t+1)} = S_\lambda \left[\boldsymbol{\alpha}_i^{(t)} + \eta \mathbf{D}^\top \left(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i^{(t)} \right) \right], \tag{2}$$

where $\eta > 0$ is a step-size and $S_\lambda[u] = \text{sign}(u) \max(|u| - \lambda, 0)$ is the soft-thresholding operator, which is applied pointwise to each entry of an input vector.

By noting that the above iteration can be seen as a sequence of affine transformations interleaved with pointwise non-linearities S_λ, it is then tempting to interpret T iterations (2) as a multilayer feed-forward neural network with shared weights between the T layers. Following such an insight, Gregor and LeCun have proposed the LISTA algorithm [25], where the parameters are learned such that the sequence approximates well the solution of the sparse coding problem (1).

Interestingly, the LISTA algorithm can also be used to train dictionaries for supervised learning tasks. This is the approach chosen in in [30, 51] for image restoration, which considers the following iterations:

$$\boldsymbol{\alpha}_i^{(t+1)} = S_\lambda \left[\boldsymbol{\alpha}_i^{(t)} + \mathbf{C}^\top \left(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i^{(t)} \right) \right], \tag{3}$$

which differs from (2) with the presence of a matrix $\mathbf{C}$ of the same size as $\mathbf{D}$. Even if the choice $\mathbf{C} = \eta\mathbf{D}$ (which recovers ISTA) is perfectly reasonable, using a different dictionary $\mathbf{C}$ has empirically shown to provide improvements in results quality [30], probably due to faster convergence of the LISTA iterations. Then, given some fixed parameters $\mathbf{C}, \mathbf{D}$, a clean estimate $\mathbf{W}\boldsymbol{\alpha}_i^{(T)}$ for each patch $\mathbf{y}_i$ is obtained by using a dictionary $\mathbf{W}$, where T is the number of LISTA steps. The reason for allowing a different dictionary $\mathbf{W}$ than $\mathbf{D}$ is to correct the potential bias due to ℓ_1-minimization.

Finally, the denoised image $\hat{\mathbf{x}}$ is reconstructed by averaging the patch estimates:

$$\hat{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{n} \mathbf{R}_i \mathbf{W} \boldsymbol{\alpha}_i^{(T)}, \tag{4}$$

where $\mathbf{R}_i$ is the linear operator that places the patch $\hat{\mathbf{x}}_i$ at position i in the image, and we assume—by neglecting border effects for simplicity—that each pixel admits the same number m of estimates.

In contrast to classical restoration techniques based on dictionary learning, the LISTA point of view enables us to learn the model parameters $\mathbf{C}, \mathbf{D}, \mathbf{W}$ in a supervised fashion. Given a training set of pairs of noisy/clean images, we remark that the estimate $\hat{\mathbf{x}}$ is obtained from a noisy image $\mathbf{y}$ by a sequence of operations that are differentiable almost everywhere, as typical neural networks with rectified linear unit activation functions. A typical loss, which we optimize by stochastic gradient descent, is then

$$\min_{\mathbf{C},\mathbf{D},\mathbf{W},\lambda} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x}\|^2 \right],$$

where $(\mathbf{x}, \mathbf{y})$ is a pair of clean/noisy images drawn from some training distribution from which we can sample, and $\hat{\mathbf{x}}(\mathbf{y})$ is the clean estimate obtained from (4), given the noisy image $\mathbf{y}$.

3.2 A Trainable Low-Rank Sparse Coding Layer

We are now in shape to introduce a trainable layer encoding both sparsity and low-rank principles.

Spatial-Spectral Representation. As shown in [10, 20], HSI patches can be well reconstructed by using only a few basis elements obtained by principal component analysis. The authors further decompose these into a Cartesian product of separate spectral and spatial dictionaries. In this paper, we adopt a slightly different approach, where we consider a single dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ in $\mathbb{R}^{m \times p}$ as in the previous section with $m = cs^2$, but each element may be seen as a matrix of size $c \times s^2$ with low-rank structure. More precisely, we enforce the following representation

$$\forall j \in 1, \ldots, p,\ \mathbf{d}_j = \text{vec}\left(\mathbf{U}_j \times \mathbf{V}_j\right), \tag{5}$$

where $\mathbf{U}_j$ is in $\mathbb{R}^{s^2 \times r}$, $\mathbf{V}_j$ is in $\mathbb{R}^{r \times c}$, r is the desired rank of the dictionary elements, and vec$(.)$ is the operator than flattens a matrix to a vector. The hyperparameter r is typically small with $r = 1, 2$ or 3. When $r = 1$, the dictionary elements are said to be separable in the spectral and spatial domains, which we found to be a too stringent condition to achieve good reconstruction in practice.

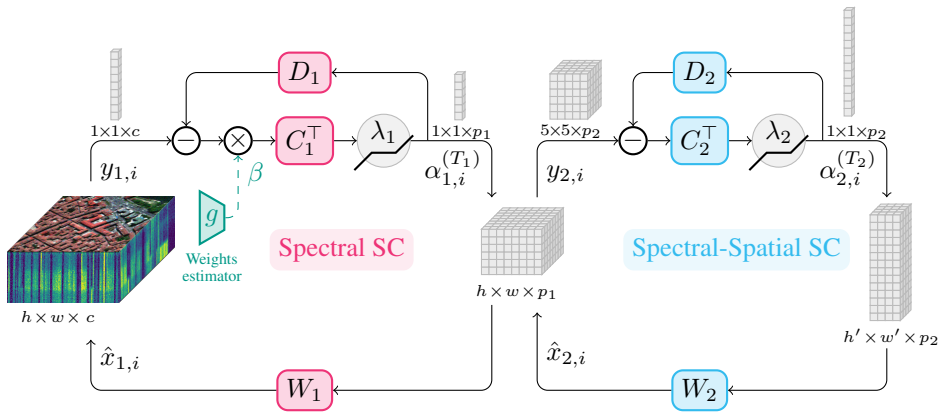


Figure 1: Proposed architecture where the first layer performs a spectral decomposition and is specific to a type of sensor, whereas the second layer may be shared for different types of sensor data.

The low-rank assumption allows us to build model with a reduced number of parameters, while encoding natural assumption about the data directly in the model architecture. Indeed, whereas a classical full-rank dictionary $\mathbf{D}$ admits cs^2p parameters, the decomposition (5) yields dictionaries with $(s^2+c)rp$ parameters only. Matrices $\mathbf{C}$ and $\mathbf{W}$ are parametrized in a similar manner.

Implementation tricks and computational complexity. Computing the sparse code $\boldsymbol{\alpha}_i^{(T)}$ by using T steps of the iteration (3) requires two matrix-vector multiplications involving $\mathbf{D}$ and $\mathbf{C}$. These operations are typically applied in parallel on all image patches by using a GPU. The complexity of the forward pass is then of order $O(nTmp)$ which is reasonable in the RGB domain since m is typically smaller than p, yielding inference times similar to that of a convolutional neural network, see [30]. Nevertheless, significant speed improvements may be achieved when a large number of channels are present and $m \gg p$. Indeed, the forward pass described in Eq (3) can be rewritten as

$$\boldsymbol{\alpha}_i^{(t+1)} = S_\lambda\left[\mathbf{G}\boldsymbol{\alpha}_i^{(t)} + \mathbf{h}\right] \qquad \text{with } \mathbf{G} = \mathbf{I} - \mathbf{C}^\top\mathbf{D} \quad \text{and} \quad \mathbf{h} = \mathbf{C}^\top\mathbf{y}_i, \tag{6}$$

where $\mathbf{G}$ is in $\mathbb{R}^{p\times p}$ and $\mathbf{h}$ is in $\mathbb{R}^p$ are precomputed at the beginning of the forward pass. The resulting complexities in terms of multiplication operations is given in the table below.

	Regular iteration (3)	$(\mathbf{G},\mathbf{h})$-parametrization (6)
Complexity (mult op.)	$2nTmp$	$p^2m + n(mp + Tp^2)$

In practice, n is typically large such that the term p^2m on the right is negligible, and we have found the implementation (6) to be an order of magnitude faster in our experiments.

Another implementation detail consists of using a different λ parameter per dictionary element, which slightly increases the number of parameters, while allowing to learn with a weighted ℓ_1-norm in (1).

3.3 The Two-Layer Sparse Coding Model with Sensor-Specific Layer

One of the main challenge in hyperspectral imaging is to train a model that can generalize to several types of sensors, which typically admit different number of spectral bands. Whereas learning a model that is tuned to a specific sensor is perfectly acceptable in many contexts, it is often useful to learn a model that is able to generalize across different types of HSI signals. To alleviate this issue, several strategies have been adopted such as (i) projecting signals onto a linear subspace of fixed dimension, with no guarantee that representations within this subspace can be comparable between different signals, or (ii) processing input data using a sliding window across the spectral domain.

In this paper, we address this issue by learning a two-layer model, presented in Figure 1, where the first layer is tuned to a specific sensor, whereas the second layer could be generic. Note that the second layer carries most of the model parameters (about $20\times$ more than in the first layer in our experiments). Formally, let us denote by $\boldsymbol{\alpha}$ in $\mathbb{R}^{p\times h\times w}$ the sparse encoding of an input tensor $\mathbf{y}$ in $\mathbb{R}^{c\times h\times w}$ after T iterations as described in (6). A sparse coding layer Φ naturally yields an encoder

and a decoder such that:

$$\Phi^{enc} : \mathbf{y} \mapsto \boldsymbol{\alpha}, \quad \text{and} \quad \Phi^{dec} : \boldsymbol{\alpha} \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathbf{R}_i \mathbf{W} \boldsymbol{\alpha}_i. \tag{7}$$

Given a noisy image $\mathbf{y}$, the denoising procedure described in the previous section with one layer can be written as

$$\hat{\mathbf{x}}(\mathbf{y}) = \Phi^{dec} \circ \Phi^{enc}(\mathbf{y}).$$

Then, a straightforward multilayer extension of the procedure may consists of stacking several several sparse coding layers $\Phi_1, \ldots, \Phi_L$ together to form a multilayer sparse coding denoising model:

$$\hat{\mathbf{x}}(\mathbf{y}) = \Phi_1^{dec} \circ \cdots \circ \Phi_L^{dec} \circ \Phi_L^{enc} \circ \cdots \circ \Phi_1^{enc}(\mathbf{y}).$$

The model we propose is composed of two layers, as shown in Figure 1. The first layer encodes spectrally the input HSI image, meaning that it operates on 1×1 patches, whereas the second layer encodes both spectrally and spatially the output of the first layer.

3.4 Leveraging Interpretability for Band-Dependent Noise.

One of the main advantage of using an interpretable model based on a sparse coding objective (1) is that it allows to encode domain knowledge within the model architecture. For instance, the Lasso problem (1) seen from a maximization a posteriori estimator implicitly assumes that the noise is i.i.d. If the noise variance is different on each spectral band, a natural modification to the model is to introduce weights and use a weighted-ℓ_2 data fitting term:

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2}\sum_{j=1}^{c} \beta_j \|\mathbf{M}_j(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i)\|^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{8}$$

where $\mathbf{M}_j$ is a linear operator that extracts band j from a given HSI signal. From a probabilistic point of view, if σ_j^2 denotes the variance of the noise for band j, we may choose the corresponding weight β_j to be proportional to $1/\sigma_j^2$. Yet, estimating accurately σ_j^2 is not always easy and we have found it more effective to simply learn a parametric function $\beta_j = g(\mathbf{M}_j \mathbf{y})$—here, a very simple CNN with three layers, see supplementary material for details—which is applied independently to each band. It it then easy to modify the LISTA iterations accordingly to take into account these weights, and learn the model parameters jointly with those of the parametric function g.

3.5 Self-Supervised Learning: Blind-Band Denoising with No Ground Truth Data

Even though acquiring limited ground truth data for a specific sensor is often feasible, it is also interesting to be able to train models with no ground truth at all, *e.g.*, for processing images without physical access to the sensor. In such an unsupervised setting, deep neural networks are typically trained for RGB images by using blind-spot denoising techniques [29], consisting of predicting pixel values given their context. Here, we propose a much simpler approach exploiting the spectral redundancy between channels. More precisely, each time we draw an image for training, we randomly mask one band (ore more), and train the model to reconstruct the missing band from the available ones. Formally, the training objective becomes

$$\min_{\mathbf{C},\mathbf{D},\mathbf{W},\lambda} \mathbb{E}_{\mathbf{x},\mathbf{y},S} \left[\sum_{j \notin S} \|\mathbf{M}_j(\hat{\mathbf{x}}_S(\mathbf{y}) - \mathbf{y})\|^2 \right], \tag{9}$$

where S is the set of bands that are visible for computing the sparse codes $\boldsymbol{\alpha}_i$, leading to a reconstructed image that we denote by $\hat{\mathbf{x}}_S(\mathbf{y})$. Formally, it would mean considering the objective (8), but replacing the sum $\sum_{j=1}^{c}$ by $\sum_{j \in S}$. This is in spirit similar to blind-spot denoising, except that bands are masked instead of pixels, making the resulting implementation much simpler.

4 Experiments

We now present various experiments to demonstrate the effectiveness of our approach.

Datasets. We evaluate our approach on two datasets with significantly different properties.

- *Washington DC Mall* is perhaps the most widely used dataset[1] for HSI denoising and consists of a high-quality image of size 1280×307 with 191 bands. Following [50], we split the image into two sub-images of size 600×307 and 480×307 for training and one sub-image of size 200×200 for testing. Even though the test image does not overlap with train images, they nevertheless share common characteristics. Interestingly, the amount of training data is very limited here.
- *ICVL* [4] consists of 204 images of size 1392×1300 with 31 bands. We used 100 images for training and 50 for testing as in [57] but with a different train/test split ensuring that similar images—*e.g.*, picture from the same scene—are not used twice.

Specific experiments were also conducted with the datasets APEX [26], Pavia[2], and Urban[3].

Normalization. Before denoising, HSI images are normalized to $[0, 1]$. For remote sensing datasets, we pre-compute the 2nd and 98th percentiles for each band, on the whole the training set. Then, normalization is performed on train and test images by clipping each band between those percentiles before applying bandwise min-max normalization, similar to [5, 34]. For the close-range dataset ICVL, we simply apply global min-max normalization as in [58, 57].

Noise patterns. We evaluate our model against different types of synthetic noise:

- *i.i.d Gaussian noise with known variance* σ^2, which is the same on all bands.
- *Gaussian noise with unknown band-dependent variance*: We consider Gaussian noise with different standard deviation σ_j for each band, which is uniformly drawn in a fixed interval. These standard deviations change from an image to the other and are unknown at test time.
- *Noise with spectrally correlated variance*: We consider Gaussian noise with standard deviation σ_j varying continuously across bands, following a Gaussian curve, see details in the appendix.
- *Stripes noise* : similar to [57], we applied additive stripes noise to 33% of bands. In those bands, 10-15% of columns are affected, meaning a value uniformly sampled in the interval $[-0.25, 0.25]$ is added to them. Morever, all bands are disturbed by Gaussian noise with noise intensity $\sigma = 25$.

Benchmarked models. We compare our approach to traditional methods such as bandwise BM3D [11] (implementation based on [40, 41]), BM4D [35], GLF [65]. We also included deep learning models in our benchmark such as HSID-CNN [60], HSI-SDeCNN [34] 3D-ADNet [50], SMDS-Net [58] and QRNN3D [57]. Results of HSID-CNN, HSI-SDeCNN and 3D-ADNet on Washington DC Mall (available in the appendix) are taken directly from the corresponding papers, as the train/test split is the same. Otherwise, the results were obtained by running the code obtained directly from the authors, except for SMDS-Net, where our implementation turned out to be slightly more effective. Note that the same architecture for our model was used in all our experiments (see Appendix).

Metrics. In order to assess the performances the previous methods, we used five different indexes widely used for HSI restoration, namely (i) Mean Peak Signal-to-Noise Ratio (MPSNR), which is the classical PSNR metric averaged across bands; (ii) Mean Structural Similarity Index Measurement (MSSIM), which is based on the SSIM metric [56]; (iii) Mean Feature Similarity Index Measurement (MFSIM) introduced in [63]: (iv) Mean ERGAS [13], and (v) Mean Spectral Angle Map (MSAM) [3]. We use MPSNR and MSSIM in the main paper and report the other metrics in the appendix.

Implementation details. We trained our network by minimizing the MSE between the groundtruth and restored images. For all datasets, we trained our models on randomly cropped volumes of size 56×56. Basic data augmentation schemes such as $90°$ rotations and vertical/horizontal flipping are performed. For testing, we split the HSI into overlapping volumes of size 56×56 which are independently denoised, before aggregating the outputs to the original size. Code and additional details about optimization, implementation, computational resources, are provided in the supplementary material.

Quantitative results on synthetic noise. We present in Table 1 the results obtained on the ICVL dataset (results on DCMall are presented in the appendix). Our method uses the vanilla model of Section 3.3 for the experiments with constant σ or correlated noise. For the blind denoising experiment with band-dependent σ or for the stripe noise experiment, we use the variant of Section 3.4, which is

[1] `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`
[2] `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`
[3] `https://rslab.ut.ac.ir/data`

Table 1: Denoising performance on ICVL with various types of noise patterns. The first two rows correspond to i.i.d. Gaussian noise with fixed σ per band. The next three rows corresponds to a noise level that depends on the band, taken uniformly on small interval. This is a blind-noise experiment since at test time, the noise level is unknown. The last two rows correspond to the scenarios with correlated σ across bands, and with stripe noise, respectively. See main text for details.

σ	Metrics	Noisy	BM3D	BM4D	GLF	SMDS	QRNN3D	Ours	Ours-SSL
5	MPSNR	34.47	46.17	48.85	51.25	50.91	43.81	**51.64**	50.48
	MSSIM	0.7618	0.9843	0.9916	0.9949	0.9944	0.9692	**0.9951**	0.9941
25	MPSNR	21.44	37.86	39.89	43.16	42.83	40.69	**43.77**	43.05
	MSSIM	0.1548	0.9269	0.9510	0.9695	0.9700	0.9570	**0.9752**	0.9721
50	MPSNR	16.03	34.22	34.22	39.26	39.25	38.95	**40.04**	39.45
	MSSIM	0.0502	0.8654	0.8654	0.9197	0.9382	0.9437	**0.9497**	0.9447
100	MPSNR	10.85	30.43	32.47	34.79	35.64	36.19	**36.27**	35.70
	MSSIM	0.0144	0.7557	0.8155	0.7982	0.8815	**0.9172**	0.9051	0.8955
[0-15]	MPSNR	33.89	45.81	45.35	**50.57**	48.23	44.46	50.17	49.48
	MSSIM	0.6386	0.9767	0.9735	**0.9948**	0.9900	0.9698	0.9938	0.9931
[0-55]	MPSNR	23.36	39.06	38.43	44.22	41.76	40.23	44.23	**44.41**
	MSSIM	0.2601	0.9231	0.9074	0.9818	0.9620	0.9442	**0.9823**	0.9790
[0-95]	MPSNR	19.06	36.17	35.55	41.43	38.94	38.19	**42.67**	42.01
	MSSIM	0.1614	0.8760	0.8540	0.9674	0.9357	0.9135	**0.9710**	0.9669
Corr.	MPSNR	28.85	42.73	42.13	47.05	45.98	41.78	**48.66**	47.48
	MSSIM	0.4740	0.9599	0.9070	0.9881	0.9835	0.9502	**0.9900**	0.9880
Strip.	MPSNR	19.31	24.24	24.33	24.47	32.88	33.87	**36.00**	23.30
	MSSIM	0.1176	0.4109	0.4177	0.4160	0.8247	0.8605	**0.8926**	0.3302

Table 2: Inference time per image on ICVL with $\sigma = 50$; SMDS, QRNN3D and Ours are using a V100 GPU; BM4D and GLF are using an Intel(R) Xeon(R) CPU E5-1630 v4 @ 3.70GHz.

	BM3D	BM4D	GLF	SMDS	QRNN3D	Ours	Ours-SSL
Inference time (s)	1677.4	2382.9	5565.1	74.3	17.9	**4.5**	44.9

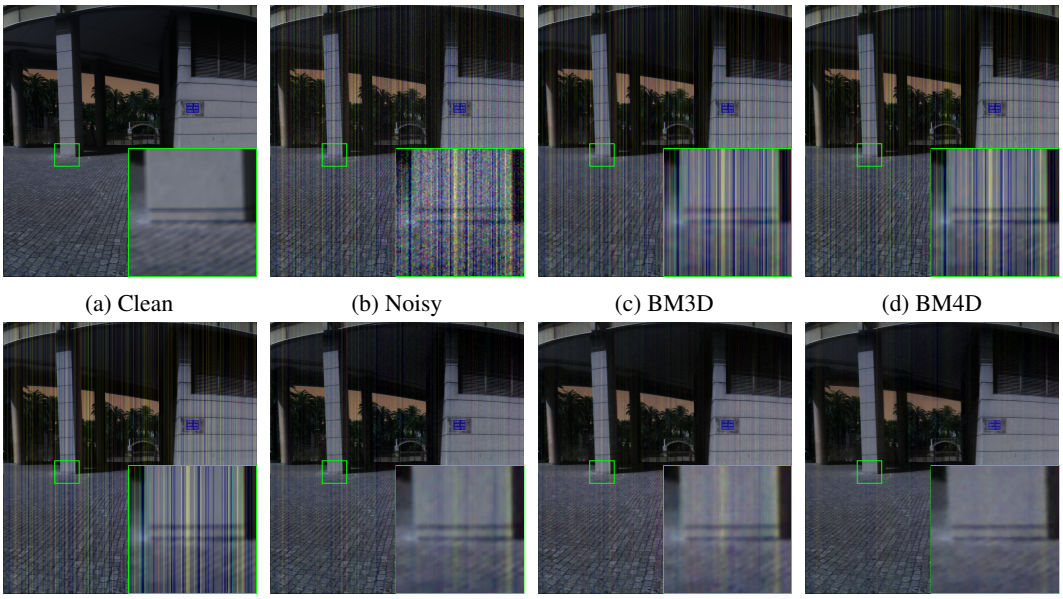
designed to deal with unknown noise level per channel. The method "Ours-SSL" implements the self-supervised learning approach of Section 3.5, which does not rely on ground-truth data.

- Our supervised approach achieves state-of-the-art results (or is close to the best performing baseline) on all settings. GLF performs remarkably well given that this baseline is learning-free.
- Our self-supervised method achieves a relatively good performance, except with stripes noise, where the model seems to overfit the noise structure. This is a limitation of the approach which is perhaps expected and overcoming this limitation would require designing a different self-supervised learning scheme; this is an interesting problem, which is beyond the scope of this paper.

A visual result on ICVL is shown in Figure 2 for stripes noise. Inference times are provided in Table 2, showing that our approach is computationally efficient.

Results on real noise. We also conducted a denoising experiment on the Urban dataset, reporting a visual result in Figure 3. Deep models were pre-trained on the APEX dataset, which has the same number of channels as Urban (even though the sensors are different), showing that denoising models trained on synthetic Gaussian noise with $\sigma = 50$ can potentially transfer well on real data.

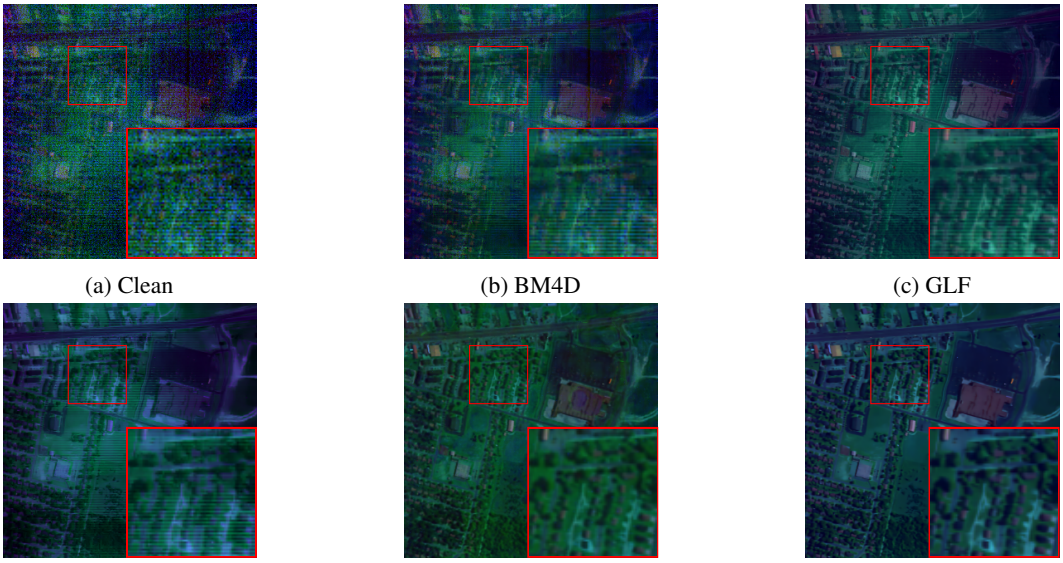
Comments on the additional results presented in the appendix. The appendix also contains (i) results on the DCMall dataset including additional baselines mentioned above; (ii) error bars for parts of our experimental results in order to assess their statistical significance; (iii) an experiment when learning simultaneously on several datasets with different types of sensors showing that the second layer can be generic and effective at the same time; (iv) additional visual results; (v) various ablation studies to illustrate the importance of different components of our method.



(a) Clean (b) Noisy (c) BM3D (d) BM4D

(e) GLF (f) SMDS-Net (g) QRNN3D (h) Ours

Figure 2: Visual results for the denoising experiment with stripes noise on ICVL with bands 9, 15, 28.



(a) Clean (b) BM4D (c) GLF

(d) SMDS-Net (e) QRNN3D (f) Ours

Figure 3: Visual result on a real HSI denoising experiment on Urban dataset with bands 1, 108, 208.

Broader Impact

Our paper addresses the problem of denoising the signal, which is a key pre-processing step before using hyperspectral signals in concrete applications. As such, it is necessarily subject to dual use. For instance, HSI may be used for environmental monitoring, forestry, yield estimation in agriculture, natural disaster management planning, astronomy, archaeology, and medicine. Yet, HSI is also used by the petroleum industry for finding new oil fields, and has obvious military applications for surveillance. We believe the potential benefits of HSI for society are large enough to outweigh the potential harm. Nevertheless, we are planning to implement appropriate dissemination strategies to mitigate the risk of misuse for this work (notably with restrictive software licenses), while targeting a gold standard regarding the scientific reproducibility of our results.

References

[1] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa. Hyperspectral imaging: A review on UAV-Based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9(11), 2017.

[2] H. Akbari, L. Halig, D. M. Schuster, B. Fei, A. Osunkoya, V. Master, P. Nieh, and G. Chen. Hyperspectral imaging and quantitative analysis for prostate cancer detection. *Journal of Biomedical Optics*, 17(7):1–11, 2012.

[3] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. M. Bruce. Comparison of pansharpening algorithms: Outcome of the 2006 grs-s data-fusion contest. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3012–3021, 2007.

[4] B. Arad and O. Ben-Shahar. Sparse recovery of hyperspectral signal from natural RGB images. In *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[5] N. Audebert, B. Saux, and S. Lefèvre. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):159–173, 2019.

[6] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research (JMLR)*, 18:1–43, 2018.

[7] J. M. Bioucas-Dias and J. M. P. Nascimento. Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445, 2008.

[8] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2):6–36, 2013.

[9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[10] A. Chakrabarti and T. Zickler. Statistics of real-world hyperspectral images. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[12] C. F. Dantas, J. E. Cohen, and R. Gribonval. Hyperspectral image denoising using dictionary learning. In *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2019.

[13] Q. Du, N. H. Younan, R. King, and V. P. Shah. On the performance evaluation of pan-sharpening techniques. *IEEE Geoscience and Remote Sensing Letters*, 4(4):518–522, 2007.

[14] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[15] G. Elmasry, M. Kamruzzaman, D.-W. Sun, and Paul Allen. Principles and applications of hyperspectral imaging in quality evaluation of agro-food products: A review. *Critical Reviews in Food Science and Nutrition*, 52(11):999–1023, 2012.

[16] H. Fan, C. Li, Y. Guo, G. Kuang, and J. Ma. Spatial–Spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 56(10):6196–6213, 2018.

[17] B. Fei. Chapter 3.6 - Hyperspectral imaging in medical applications. In J. M. Amigo, editor, *Hyperspectral Imaging*, volume 32 of *Data Handling in Science and Technology*, pages 523–565. Elsevier, 2020.

[18] Y.-Z. Feng and Da-Wen Sun. Application of hyperspectral imaging in food safety inspection and control: A review. *Critical Reviews in Food Science and Nutrition*, 52(11):1039–1058, 2012.

[19] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, 2003.

[20] Y. Fu, A. Lam, I. Sato, and Y. Sato. Adaptive spatial-spectral dictionary learning for hyperspectral image denoising. In *Proc. International Conference on Computer Vision (ICCV)*, 2015.

[21] A. F. Goetz. Three decades of hyperspectral remote sensing of the Earth: A personal view. *Remote Sensing of Environment*, 113:S5–S16, 2009.

[22] X. Gong, W. Chen, and J. Chen. A low-rank tensor dictionary learning method for hyperspectral image denoising. *IEEE Transactions on Signal Processing*, 68:1168–1180, 2020.

[23] C. A. G. Gonzalez, O. Absil, and M. van Droogenbroeck. Supervised detection of exoplanets in high-contrast imaging sequences. *Astronomy & Astrophysics*, 613:A71, May 2018.

[24] A. A. Gowen, Y. Feng, E. Gaston, and V. Valdramidis. Recent applications of hyperspectral imaging in microbiology. *Talanta*, 137:43–54, 2015.

[25] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proc. International Conference on Machine Learning (ICML)*, 2010.

[26] K. I. Itten, F. Dell'Endice, A. Hueni, M. Kneubühler, D. Schläpfer, D. Odermatt, F. Seidel, S. Huber, J. Schopfer, T. Kellenberger, et al. Apex-the hyperspectral esa airborne prism experiment. *Sensors*, 8(10):6235–6259, 2008.

[27] J. Kerekes and J. Baum. Hyperspectral imaging system modeling. *Lincoln Laboratory Journal*, 14(1):117–130, 2003.

[28] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE signal processing magazine*, 19(1):44–57, 2002.

[29] S. Laine, T. Karras, J. Lehtinen, and T. Aila. High-quality self-supervised deep image denoising. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2019.

[30] B. Lecouat, J. Ponce, and J. Mairal. Fully trainable and interpretable non-local sparse models for image restoration. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[31] Y. Liu, H. Pu, and D.-W. Sun. Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications. *Trends in Food Science & Technology*, 69:25–35, 2017.

[32] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang. Recent advances of hyperspectral imaging technology and applications in agriculture. *Remote Sensing*, 12(16), 2020.

[33] G. Lu and B. Fei. Medical hyperspectral imaging: A review. *Journal of Biomedical Optics*, 19(1):1–24, 2014.

[34] A. Maffei, J. M. Haut, M. E. Paoletti, J. Plaza, L. Bruzzone, and A. Plaza. A single model CNN for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2516–2529, 2020.

[35] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, 2013.

[36] S. Mahesh, D. Jayas, J. Paliwal, and N. White. Hyperspectral imaging to classify and monitor quality of agricultural materials. *Journal of Stored Products Research*, 61:17–26, 2015.

[37] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):791–804, 2012.

[38] J. Mairal, F. Bach, J. Ponce, et al. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[39] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.

[40] Y. Mäkinen, L. Azzari, and A. Foi. Exact transform-domain noise variance for collaborative filtering of stationary correlated noise. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

[41] Y. Mäkinen, L. Azzari, and A. Foi. Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching. *IEEE Transactions on Image Processing*, 29:8339–8354, 2020.

[42] D. G. Manolakis, R. B. Lockwood, and T. W. Cooley. *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge University Press, Cambridge, 2016.

[43] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[44] H. Othman and S.-E. Qian. Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage. *IEEE Transactions on Geoscience and Remote Sensing*, 44(2):397–408, 2006.

[45] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[46] N. Qi, Y. Shi, X. Sun, and B. Yin. TenSR: Multi-dimensional tensor sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[47] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot. Noise Reduction in Hyperspectral Imagery: Overview and Application. *Remote Sensing*, 10(3):482, Mar. 2018.

[48] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson. Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2458–2467, 2014.

[49] B. Rasti, M. O. Ulfarsson, and P. Ghamisi. Automatic hyperspectral image restoration using sparse and low-rank modeling. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2335–2339, 2017.

[50] Q. Shi, X. Tang, T. Yang, R. Liu, and L. Zhang. Hyperspectral image denoising using a 3-D attention denoising network. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[51] D. Simon and M. Elad. Rethinking the CSC model for natural images. In *Adv. Neural Information Processing Systems (NeurIPS)*, 2019.

[52] V. Studer, J. Bobin, M. Chahid, H. S. Mousavi, E. Candes, and M. Dahan. Compressive fluorescence microscopy for biological and hyperspectral imaging. *Proceedings of the National Academy of Sciences*, 109(26):E1679–E1687, 2012.

[53] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[54] M. Wang, Q. Wang, J. Chanussot, and D. Hong. L_0-l_1 hybrid total variation regularization and its applications on hyperspectral image mixed noise removal and compressed sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[55] M. Wang, Q. Wang, J. Chanussot, and D. Li. Hyperspectral image mixed noise removal based on multidirectional low-rank modeling and spatial-spectral total variation. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1):488–507, 2021.

[56] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[57] K. Wei, Y. Fu, and H. Huang. 3-D quasi-recurrent neural network for hyperspectral image denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[58] F. Xiong, J. Zhou, J. Lu, and Y. Qian. SMDS-Net: Model Guided Spectral-Spatial Network for Hyperspectral Image Denoising. *arXiv:2012.01829*, 2020.

[59] Q. Yuan, L. Zhang, and H. Shen. Hyperspectral image denoising with a Spatial–Spectral view fusion strategy. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2314–2325, 2014.

[60] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang. Hyperspectral Image Denoising Employing a Spatial–Spectral Deep Residual Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1205–1218, Feb. 2019.

[61] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4729–4743, 2014.

[62] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[63] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

[64] Y.-Q. Zhao and J. Yang. Hyperspectral image denoising via sparse representation and low-rank constraint. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):296–308, 2015.

[65] L. Zhuang and J. M. Bioucas-Dias. Hyperspectral image denoising based on global and non-local low-rank factorizations. In *IEEE International Conference on Image Processing (ICIP)*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See broader impact section
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]
(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] see Supplementary material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see Supplementary material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] To some extent. This study is conducted for our approaches and some baselines and is provided in the supplementary material. It was however not conducted for all experiments since it would have greatly increased the amount of computational resources required for this work.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see Supplementary material

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We use publicly available code and datasets whose license is compatible with research purposes.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]