# Feature Extraction for Hyperspectral Imagery

*The evolution from shallow to deep: Overview and toolbox*

**BEHNOOD RASTI, DANFENG HONG, RENLONG HANG, PEDRAM GHAMISI, XUDONG KANG, JOCELYN CHANUSSOT, AND JON ATLI BENEDIKTSSON**

xxxxx

Hyperspectral images (HSIs) provide detailed spectral information through hundreds of (narrow) spectral channels (also known as *dimensionality* or *bands*), which can be used to accurately classify diverse materials of interest. The increased dimensionality of such data makes it possible to significantly improve data information content but provides a challenge to conventional techniques (the so-called curse of dimensionality) for accurate analysis of HSIs.

Feature extraction (FE), a vibrant field of research in the hyperspectral community, evolved through decades of research to address this issue and extract informative features suitable for data representation and classification. The advances in FE were inspired by two fields of research—the popularization of image and signal processing along with machine (deep) learning—leading to two types of FE approaches: the shallow and deep techniques. This article outlines the advances in these approaches for HSI by providing a technical overview of state-of-the-art techniques, offering useful entry points for researchers at different levels (including students, researchers, and senior researchers) willing to explore novel investigations on this challenging topic.

In more detail, this article provides a bird's eye view of shallow [both supervised FE (SFE) and unsupervised FE (UFE)] and deep FE approaches, with a specific focus on hyperspectral FE and its application to HSI classification. Additionally, this article compares 15 advanced techniques with an emphasis on their methodological foundations and classification accuracies. Furthermore, to push this vibrant field of research forward, an impressive amount of code and libraries are shared on GitHub, which can be found in [131].

*Digital Object Identifier 10.1109/MGRS.2020.2979764*
*Date of current version: 24 April 2020*

## A BRIEF INTRODUCTION ON FE

HSI technology provides detailed spectral information by sampling the reflective portion of the electromagnetic spectrum, covering a wide range, from the visible region (0.4–0.7 m) to the short-wave infrared region (almost 2.4 m). Hyperspectral sensors can also characterize the emissive properties of objects by acquiring data in the range of the midwave and long-wave infrared regions, in hundreds of narrow, contiguous spectral channels.

Detailed spectral information provided by hyperspectral sensors presents both challenges and opportunities. For instance, HSIs can be used to differentiate between different classes of interest with slightly different spectral characteristics [1]. However, most of the commonly used methods utilized for the analysis of gray scale, color, or multispectral images cannot be extended to analyze HSIs for several reasons, as detailed in the "Unique Properties of High-Dimensional Data" section.

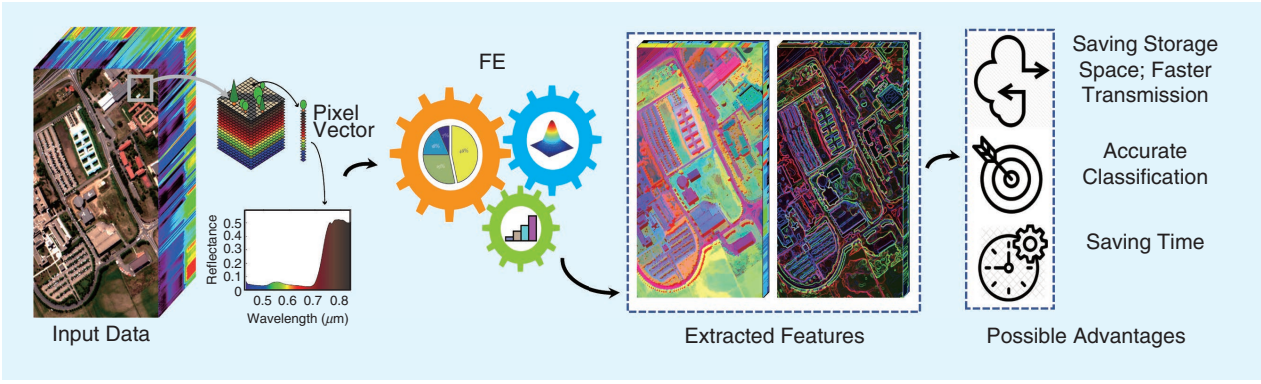


**FIGURE 1.** The FE technique and its advantages for HSI analysis.

The limited availability of training samples (a common issue in remote sensing) dramatically impacts the performances of supervised classification approaches due to the high dimensionality of HSIs, which poses a problem for designing robust statistical estimations. FE can be used to address this. It can be described as finding a set of vectors that represent an observation while reducing the dimensionality by transforming the input data linearly or nonlinearly to another domain, thereby extracting informative features in the new domain. The use of FE techniques can be advantageous for a number of reasons, which are illustrated in Figure 1 and described in the following sections.

### *UNIQUE PROPERTIES OF HIGH-DIMENSIONAL DATA*

Several studies (e.g., [2]–[4]) have demonstrated the unique geometrical, statistical, and asymptotic properties of high-dimensional data compared with red, green, blue (RGB) and multispectral images. These properties, which have been shown through experimental and theoretical examples, explain why most analytical approaches developed for RGB and multispectral images are not applicable to HSIs [5]. Among those experimental examples, we can recall that 1) as dimensionality increases, the volume of a hypercube concentrates in corners, or 2) as dimensionality increases, the volume of a hypersphere concentrates in an outside shell. With respect to these examples, the following conclusions have been drawn.

- A high-dimensional feature space is almost empty, which indicates that multivariate data in $\mathbb{R}^p$ ($p$ represents the number of bands, spectral channels, or dimensions) can usually be represented in a lower-dimensional space (referred to as *subspace*) without losing considerable information in terms of class separability [5].
- Since the high-dimensional feature space is almost empty (i.e., Gaussian distributed data have a tendency to concentrate in the tails, whereas uniformly distributed data have a tendency to be concentrated in the corners), the density estimation of hyperspectral data for both Gaussian and uniform distributions becomes extremely challenging.

Fukunaga [6] claimed that there is a relation between the type of classifier, required number of training samples, and number of input dimensions. As reported in [6], the required number of training samples is linearly related to the dimensionality for linear classifiers and to the square of the dimensionality for quadratic classifiers (e.g., the Gaussian maximum likelihood classifier [6]); for nonparametric classifiers, the number of required samples exponentially increases as the dimensionality increases. Landgrebe [7] showed the groundbreaking fact that too many spectral bands might have negative impacts in terms of expected classification performance.

When dimensionality increases, with a constant and limited number of training samples, more statistics must be estimated. Thus, the accuracy of the statistical estimation decreases, although higher spectral dimensions increase the separability between the classes. This leads to a decrease in classification accuracies beyond an unknown number of bands. These problems are related to the curse of dimensionality, also known as the *Hughes phenomenon* [8]. This finding went against the general understanding of hyperspectral data, which wrongly maintained that full dimensionality is always better than subspace in terms of classification accuracies.

The unique characteristics of high-dimensional data, as discussed, have a pronounced impact on the performances of supervised classifiers [9], as they demand an adequate number of training samples, which is almost impossible to obtain in practice since the collection of such training samples is either expensive or time demanding. To address this issue, FE-based dimensionality reduction is found to be effective.

### *STORAGE SYSTEMS AND PROCESSING TIMES*

We are now in the era of massive data acquisition. Statistics demonstrate that the cumulative volume of existing big data has increased tremendously, from 4.4 ZB in 2013 to 44 ZB in 2020 [130]. The Earth observation (EO) community has also faced a similar trend because of the enormous volume and variety of data being generated by EO missions. For example, EnMAP, a hyperspectral satellite mission, is planning to capture hyperspectral data with a maximum ground coverage of 5,000 × 30 km per day and a target revisit time of four days (±30°) with 512-Gb onboard mass memory [10]. FE-based dimensionality reduction helps in

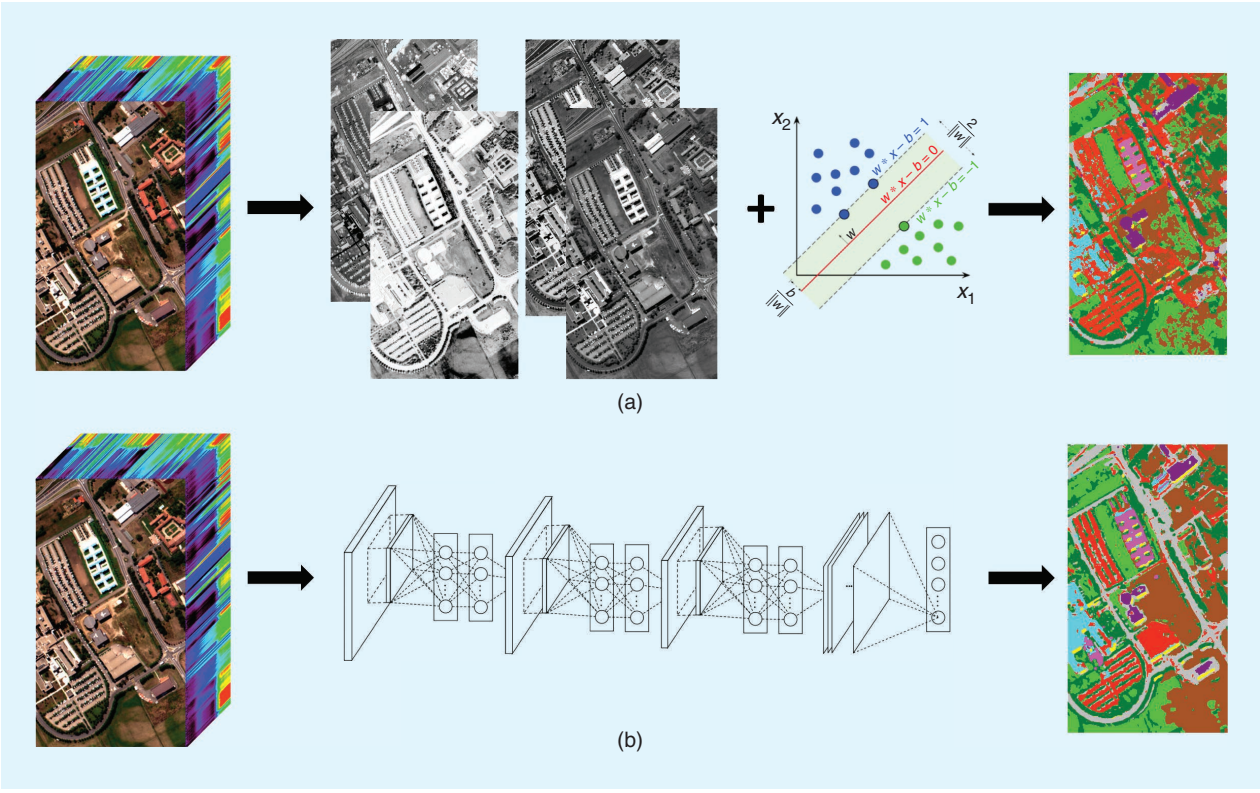


**FIGURE 2.** FE via (a) machine learning and (b) DL.

data compression, which leads to faster transmission time, removal of redundant features, smaller storage space requirements, and decreasing the required time for performing the same computations.

### *MACHINE LEARNING AND FE: AN EVER-GROWING RELATION*

Figure 2(a) illustrates the basic idea of a machine-learning approach, which consists of FE and classification. In machine learning, users are requested to provide guidelines for the machine (algorithm). This is usually done by applying handcrafted FE approaches to produce informative features for the subsequent classifier. At the very beginning, each image pixel is regarded as a pattern, and its spectrum (i.e., a vector of different values of a pixel in different spectral channels) is considered the initial set of features. This set of features, also known as *spectral features*, suffers from two important downsides: 1) the features are often redundant, and 2) they do not consider the spatial dependencies of the adjacent pixels.

To address the first issue, a feature-reduction step (through FE or selection) can be applied to reduce the dimensionality of the input data (from $p_1$ dimensions in the original data to $p_2$ dimensions in a new feature space, $p_2 < p_1$). This step, also called *spectral FE*, tries to preserve the key spectral information of the data by reducing the dimensionality and maximizing separability between classes. It is interesting that the second issue can also be addressed using FE approaches. Note that, here, the FE step, also known as *spatial FE*, is not aimed at reducing the dimensionality; instead, it is intended to model (extract) spatial contextual information suitable for the subsequent classification or object-detection step, and it usually leads to an increase in the number of features. The simultaneous use of spectral and spatial features for hyperspectral data classification has been studied in numerous works, such as [5] and [11]–[13].

Deep learning (DL), as shown in Figure 2(b), which is regarded as a subset of machine learning, tries to automatize the building blocks of machine-learning approaches (i.e., FE and classification) by developing an end-to-end framework that takes the input, performs automatic FE and classification by considering the unique nature of the input data (instead of those handcrafted FE designs in machine learning), and outputs classification maps. It turns out that, if an adequate amount of training data is supplied, DL approaches can outperform any other shallow machine-learning approaches in terms of accuracy. Here, a question arises: Due to the fact that, in the remote sensing community, the available training data are often limited, would advanced DL-based approaches outperform their shallow alternatives in terms of accuracy? This issue is addressed in this article.

Based on the previous descriptions, FE is a key step in both machine learning and DL—a concept that has evolved significantly through time from unsupervised to (semi-) supervised, from spectral or spatial to spectral and spatial, from manual to automatic, from handcrafted to end-to-end, and from shallow to deep.

### *CONTRIBUTIONS*

This article provides a detailed and organized overview of hyperspectral FE techniques, categorized into two general sections: shallow (further divided into supervised and unsupervised) and deep. Each section provides a critical overview of the state of the art, which is mainly rooted in the signal and image processing, statistical inference, and machine- (deep-) learning fields. Then, a few representative and advanced FE approaches are chosen from each of these categories for further analysis and comparisons (mostly in terms of usefulness for classification).

This article, therefore, contributes to answering the following questions:

- When it comes to hyperspectral data in EO, are DL-based FE approaches better alternatives than their conventional (yet advanced) shallow FE techniques?
- Which factors should be considered to design robust shallow and deep FE techniques?

In addition, to further promote this field of research, the article is accompanied by a significant amount of code and libraries for hyperspectral FE, made publicly available on GitHub, which can be found in [131].

Finally, several possible future directions are highlighted. To make the contribution of this article clearer, here, we briefly discuss the related existing literature. The work of Li et al. [14] is dedicated to the evolution of discriminant-analysis-based FE models, a specific type of dimensionality-reduction approach. Jia et al. [15] reviewed FE and data mining works, mostly published in 2012 and earlier. Since 2012, however, many deep and shallow FE approaches have been developed, and these are critically reviewed and compared against each other in this article. Sun and Du [16] focused only on feature selection approaches, whereas our article covers FE techniques; therefore, they complement each other.

## DATA SETS AND NOTATION

### *DATA SETS*

#### INDIAN PINES 2010

This data set (Figure 3) is a very-high-resolution (VHR) HSI acquired by the ProSpecTIR system over an area near Purdue University, Indiana, on 24 and 25 May 2010. In this article, we use a subset of $445 \times 750$ pixels with 360 spectral bands. The data set has a spatial resolution of 2 m and spectral width of 5 nm, and it contains the 16 land cover classes listed in Table 1, which also shows the training and test sets used in this study. Table 1 gives the number of samples, including training and test samples, used in the experimental section.

#### HOUSTON UNIVERSITY 2013

This data set was acquired on 23 June 2012 by the Compact Airborne Spectrographic Imager (CASI) over the campus of the University of Houston and the neighboring urban area [132]. The average height of the sensor was 5,500 ft. The data contain $349 \times 1{,}905$ pixels, with a spatial resolution of 2.5 m, and 144 spectral bands ranging from 0.38 to 1.05 $\mu$m. The data set includes 15 classes of interest, shown in Figure 4. A color composite representation of the data and the corresponding training and test samples used in this study are also shown in Figure 4. The number of training and test samples for different classes of interest used in the experiments is given in Table 2.

#### HOUSTON UNIVERSITY 2018

This data set was acquired on 16 February 2017 by the hyperspectral imager CASI 1500 over the area of the University of Houston. In this article, we utilized the training portion of the whole data set, which was distributed by the

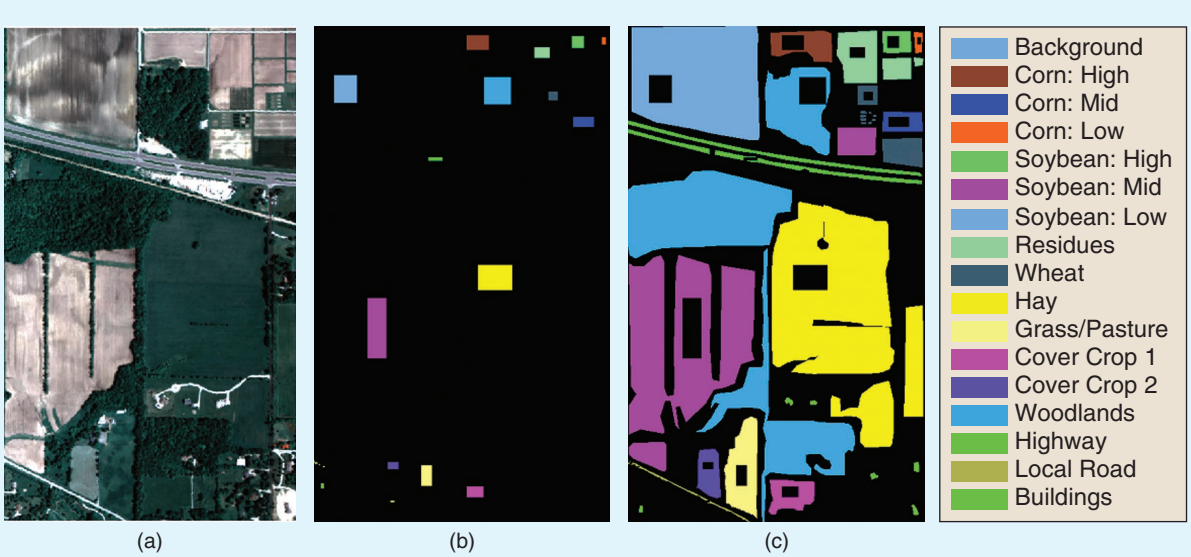


**FIGURE 3.** The Indian Pines 2010 data set: the (a) RGB composition, (b) training set, and (c) test set.

**TABLE 1. THE INDIAN PINES 2010 DATA SET: THE NUMBER OF TRAINING AND TEST SAMPLES AND THE TOTAL NUMBER OF SAMPLES PER CLASS.**

| CLASS NUMBER | CLASS NAME | TRAINING SAMPLES | TEST SAMPLES | SAMPLES |
|---|---|---|---|---|
| 1 | Corn: high | 726 | 2,661 | 3,387 |
| 2 | Corn: mid | 465 | 1,275 | 1,740 |
| 3 | Corn: low | 66 | 290 | 356 |
| 4 | Soybean: high | 324 | 1,041 | 1,365 |
| 5 | Soybean: mid | 2,548 | 35,317 | 37,865 |
| 6 | Soybean: low | 1,428 | 27,782 | 29,210 |
| 7 | Residues | 368 | 5,427 | 5,795 |
| 8 | Wheat | 182 | 3,205 | 3,387 |
| 9 | Hay | 1,938 | 48,107 | 50,045 |
| 10 | Grass/pasture | 496 | 5,048 | 5,544 |
| 11 | Cover crop 1 | 400 | 2,346 | 2,746 |
| 12 | Cover crop 2 | 176 | 1,988 | 2,164 |
| 13 | Woodlands | 1,640 | 46,919 | 48,559 |
| 14 | Highway | 105 | 4,758 | 4,863 |
| 15 | Local road | 52 | 450 | 502 |
| 16 | Buildings | 40 | 506 | 546 |
| Total | | 10,954 | 187,120 | 198,074 |

Image Analysis and Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society (GRSS) and the University of Houston for the 2018 data fusion contest [133], [134]. The data cover the spectral range of 380 to 1,050 nm with 48 bands and a ground-sampling distance of 1 m. The data set contains 601 × 2,384 pixels and 20 land cover classes of interest, shown in Figure 5. The VHR RGB image is downsampled (Figure 5), together with the corresponding training and test samples used in this study. The number of training and test samples for different classes of interest used in the experiments is given in Table 3.

### *NOTATION*

The observed HSI is denoted by $\mathbf{X} \in \mathbb{R}^{p \times n}$, where $p$ and $n$ are the number of spectral bands and pixels in each band, respectively; $d$ indicates the dimension of the feature space (the subspace); $\mathbf{X}_m \in \mathbb{R}^{p \times m}$, where $m < n$ denotes the matrix, which contains the training samples; $\mathbf{y}_m \in \mathbb{R}^{1 \times m}$ denotes the vector, which contains the class labels, where $y_i \in \{1, 2, \ldots, k\}$; and $k$ denotes the number of classes. $\mathbf{I}$ is the identity matrix, and $\hat{\mathbf{X}}$ is the estimate of matrix $\mathbf{X}$. The Frobenius norm is denoted by $\|\cdot\|_F$, and $\mathrm{tr}(\mathbf{X})$ denotes the

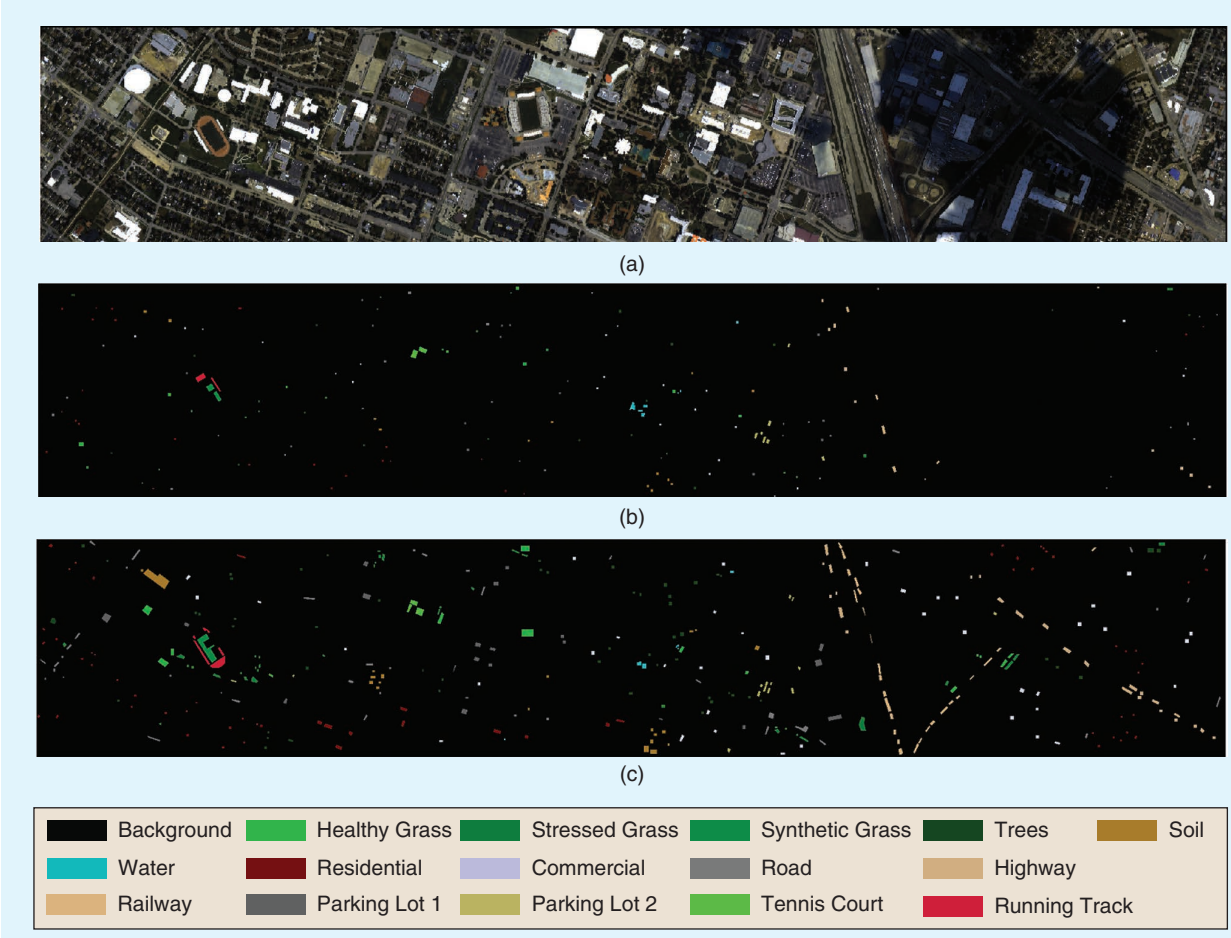


**FIGURE 4.** The Houston University 2013 data set: the (a) RGB composition, (b) training set, and (c) test set.

trace of matrix $\mathbf{X}$. The definitions of the symbols used in the article are given in Table 4.

## SHALLOW FE TECHNIQUES

### *UFE TECHNIQUES*

*UFE* often refers to FE techniques that do not incorporate the knowledge of the ground (ground reference or labeled samples) to extract features. UFE techniques often rely on intrinsic characteristics of the HSI data, such as geometric, spatial, or spectral information, to extract the features. Arguably, the main advantage of UFE, compared with other FE techniques, is the lack of need for training samples, which is of great importance in the case of remote sensing data sets. In this article, four major UFE groups widely used for HSI analysis are studied and categorized in the following sections. Figure 6 provides graphical abstracts of those groups.

Before explaining UFE techniques in more detail, we briefly refer to three groups of commonly used FE techniques that could also be considered UFE but are not studied in depth in this article due to their specific applications. The first group includes a range of approaches, such as normalized differential vegetation index and normalized differential water index, that often rely on the knowledge of the characteristics of the sensors. The second group includes unmixing techniques, which could be assumed to be UFE techniques. These often exploit optimization techniques to show the fractions of materials existing in pixels based on some assumptions on the spectral signatures of the materials. Therefore, the final features extracted represent different materials in the scene at the subpixel level [17]. The third group includes an impressive number of approaches based on mathematical morphology that hierarchically extract spatial and contextual information from the input image, which usually leads to a significant increase in the number of features [18].

#### CONVENTIONAL DATA PROJECTION/ TRANSFORMATION TECHNIQUES

Numerous UFE techniques fall into this category. The conventional techniques categorized in this group are often designed to linearly project or transform the data, $\mathbf{X}$, in a lower-dimensional feature space (also called *subspace*), exploiting different nonlocal intrinsic characteristics of the hyperspectral data set. The transformation can be given by

$$\mathbf{Z} = \mathbf{V}^T\mathbf{X}, \tag{1}$$

where $\mathbf{Z}$ is the projected data in the lower-dimensional space and $\mathbf{V}$ is the transformation matrix or the bases for the subspace. Arguably, principal component analysis (PCA) [19] is the most conventional UFE technique, and it has been widely used for hyperspectral analysis [20]. PCA captures the maximum variance of the signal by projecting the signal on the eigenvectors of the covariance matrix $(\mathbf{C})$ using

$$\max_{\mathbf{V}} \frac{\mathbf{V}^T\mathbf{C}\mathbf{V}}{\mathbf{V}^T\mathbf{V}}. \tag{2}$$

**TABLE 2. THE HOUSTON UNIVERSITY 2013 DATA SET: THE NUMBER OF TRAINING AND TEST SAMPLES AND THE TOTAL NUMBER OF SAMPLES PER CLASS.**

| CLASS NUMBER | CLASS NAME | TRAINING SAMPLES | TEST SAMPLES | SAMPLES |
|---|---|---|---|---|
| 1 | Healthy grass | 198 | 1,053 | 1,251 |
| 2 | Stressed grass | 190 | 1,064 | 1,254 |
| 3 | Synthetic grass | 192 | 505 | 697 |
| 4 | Trees | 188 | 1,056 | 1,244 |
| 5 | Soil | 186 | 1,056 | 1,242 |
| 6 | Water | 182 | 143 | 325 |
| 7 | Residential | 196 | 1,072 | 1,268 |
| 8 | Commercial | 191 | 1,053 | 1,244 |
| 9 | Road | 193 | 1,059 | 1,252 |
| 10 | Highway | 191 | 1,036 | 1,227 |
| 11 | Railway | 181 | 1,054 | 1,235 |
| 12 | Parking lot 1 | 192 | 1,041 | 1,233 |
| 13 | Parking lot 2 | 184 | 285 | 469 |
| 14 | Tennis court | 181 | 247 | 428 |
| 15 | Running track | 187 | 473 | 660 |
| Total | | 2,832 | 12,197 | 15,029 |

A widely used HSI UFE techniques is the maximum noise fraction (MNF) [21] or noise-adjusted principal components [22]; this technique seeks a projection in which the signal-to-noise ratio (SNR) is maximized. MNF uses the following optimization:

$$\max_{\mathbf{V}} \frac{\mathbf{V}^T\mathbf{C}\mathbf{V}}{\mathbf{V}^T\mathbf{C}_n\mathbf{V}}, \tag{3}$$

where $\mathbf{C}_n$ is the noise covariance matrix. Another conventional technique is independent component analysis (ICA) [23]. ICA assumes a linear mixture model of the non-Gaussian independent source signals and the mixing matrix, both of which are simultaneously estimated; therefore, ICA is referred to as *blind source separation*. ICA is also often used for HSI analysis [24].

To cope with the nonlinearity of HSI data, the kernel (nonlinear) versions of the aforementioned techniques, i.e., kernel MNF [25], kernel ICA (KICA) [26], and kernel PCA (KPCA) [27], have been proposed. Using the kernel trick, the data are projected into a feature space where the inner products are defined using a kernel function. KICA and KPCA were used as UFE techniques for change detection and classification in [28] and [29], respectively. In [30], discrete wavelet transformation (DWT) was used for hyperspectral FE. Since DWT does not reduce the dimension, in [30], linear discriminant analysis (LDA) was exploited to reduce the dimension.

#### BAND-CLUSTERING/-SPLITTING AND MERGING-BASED TECHNIQUES

Figure 6(b) shows the basic steps of band-clustering and merging-based FE methods. As indicated, the core idea behind this group of methods is to split the spectral bands

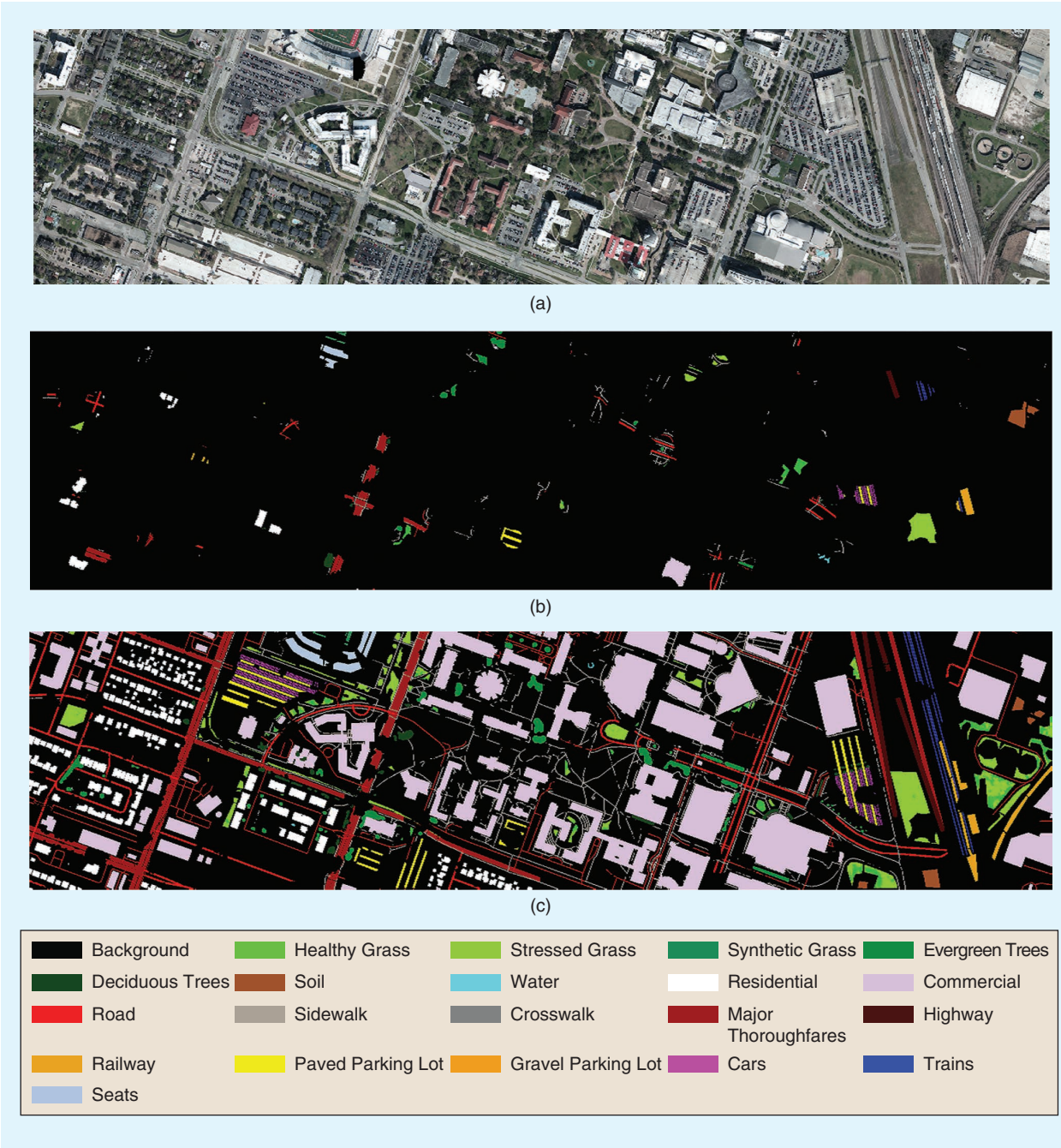


**FIGURE 5.** The Houston University 2018 data set: the (a) VHR RGB image (downsampled), (b) training set, and (c) test set.

into several groups in which the spectral bands have very high correlation. Hence, the proposed techniques often use similarity and dissimilarity criteria to split the spectral bands into several nonoverlapping groups. By selecting or fusing the bands of each group, some representative bands or features of different groups are obtained. Furthermore, followed by the merging step, some band-filtering and processing operations can be also used to further improve the discrimination of the resulting features. This group of techniques is often computationally cheap and, thus, is often used in real applications. On the other hand, spectral information is often neglected by the methods in this category.

For band clustering and merging, two algorithms are proposed in [31]. The first selects discriminative bases by considering all of the classes simultaneously; however, the second selects the best bases for a pair of classes at a time. In [32], a hierarchical clustering algorithm was introduced to split and cluster the hyperspectral bands, where the representative band for each cluster is selected based on both a mutual information (MI) criterion and a divergence-based criterion. Another band-clustering technique was proposed

in [33], where the splitting is done by minimizing an MI criterion iteratively applied on averaged bands. Iterative algorithms were proposed in [34] for both splitting and merging the bands. The splitting procedure is done using the Pearson correlation coefficient between adjacent bands; then, the merging is applied by averaging over the split bands.

Besides splitting/clustering and merging hyperspectral bands, another operation is to further improve the feature discrimination by band filtering or processing. For example, a hyperspectral FE using image fusion and recursive filtering was given in [35], where the adjacent bands are fused by averaging, and, then, recursive filtering was used to extract spatial information. In [36], intrinsic image decomposition was applied for processing the merged bands, which can effectively remove information that is not related to the material of different objects. After that, multiple improved versions of intrinsic decomposition-based band-processing methods were developed [37], [38]. In [39], a relative total-variation-based structure-extraction method was applied for band processing so as to construct multiscale structural features that are robust to image noise.

### LOW-RANK RECONSTRUCTION-BASED TECHNIQUES

Low-rank reconstruction-based FE techniques proposed by Rasti et al. [40]–[43] are based on finding an orthogonal subspace by minimizing a constrained cost function. They exploit low-rank models and reconstruction-based optimization frameworks to extract features. The optimization frameworks take into account prior knowledge of the data using different types of penalties. Due to the noise assumption in the low-rank model used, this group of FE techniques is robust to noise. They are often computationally expensive compared to groups 1 and 2 due to the iterative algorithms used to solve the (nonconvex) optimization problem.

Wavelet-based sparse reduced rank regression (WSRRR) [41] applies the sparsity prior on the wavelet coefficients, considering that the projected data on wavelet bases are sparse. WSRRR uses the model

$$\mathbf{X} = \mathbf{V}^T\mathbf{Q}\mathbf{D}_2 + \mathbf{N}, \tag{4}$$

where $\mathbf{D}_2$ represents 2D wavelet bases, $\mathbf{X}$ is the observed HSI, $\mathbf{V}$ contains the orthogonal subspace bases, and $\mathbf{N}$ is the noise and model error. WSRRR simultaneously estimates the low-rank projection matrix and the wavelet coefficients $\mathbf{W}$, which minimizes

$$(\hat{\mathbf{V}}, \hat{\mathbf{Q}}) = \underset{\mathbf{Q},\mathbf{V}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{X} - \mathbf{V}^T\mathbf{Q}\mathbf{D}_2\|_F^2 + \sum_{j=1}^{d} \lambda_j \|\mathbf{q}_{(j)}^T\|_1 \quad \text{s.t.} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}. \tag{5}$$

Note that the extracted features are given by $\hat{\mathbf{F}} = \hat{\mathbf{Q}}\mathbf{D}_2$.

To capture the spatial (neighboring) information, orthogonal total variation (TV) component analysis (OTVCA) was proposed in [42], where the HSI is modeled as

$$\mathbf{X} = \mathbf{V}^T\mathbf{F} + \mathbf{N}, \tag{6}$$

with matrix $\mathbf{F}$ containing the unknown features. OTVCA assumes that the hyperspectral features are spatially piecewise smooth and, therefore, exploits the TV penalty and simultaneously estimates $\mathbf{F}$ and $\mathbf{V}$ using

$$\hat{\mathbf{F}} = \underset{\mathbf{F}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{X} - \mathbf{V}^T\mathbf{F}\|_F^2 + \lambda \sum_{j=1}^{d} \mathrm{TV}(\mathbf{f}_{(j)}^T), \tag{7}$$

**TABLE 3 THE HOUSTON UNIVERSITY 2018 DATA SET: THE NUMBER OF TRAINING AND TEST SAMPLES AND THE TOTAL NUMBER OF SAMPLES PER CLASS.**

| CLASS NUMBER | CLASS NAME | TRAINING SAMPLES | TEST SAMPLES | SAMPLE |
|---|---|---|---|---|
| 1 | Healthy grass | 1,458 | 8,341 | 9,799 |
| 2 | Stressed grass | 4,316 | 28,186 | 32,502 |
| 3 | Synthetic grass | 331 | 353 | 684 |
| 4 | Evergreen trees | 2,005 | 11,583 | 13,588 |
| 5 | Deciduous trees | 676 | 4,372 | 5,048 |
| 6 | Soil | 1,757 | 2,759 | 4,516 |
| 7 | Water | 147 | 119 | 266 |
| 8 | Residential | 3,809 | 35,953 | 39,762 |
| 9 | Commercial | 2,789 | 220,895 | 223,684 |
| 10 | Road | 3,188 | 42,622 | 45,810 |
| 11 | Sidewalk | 2,699 | 31,303 | 34,002 |
| 12 | Crosswalk | 225 | 1,291 | 1,516 |
| 13 | Major thoroughfares | 5,193 | 41,165 | 46,358 |
| 14 | Highway | 700 | 9,149 | 9,849 |
| 15 | Railway | 1,224 | 5,713 | 6,937 |
| 16 | Paved parking lot | 1,179 | 10,296 | 11,475 |
| 17 | Gravel parking lot | 127 | 22 | 149 |
| 18 | Cars | 848 | 5,730 | 6,578 |
| 19 | Trains | 493 | 4,872 | 5,365 |
| 20 | Seats | 1,313 | 5,511 | 6,824 |
| Total | | 34,477 | 470,235 | 504,712 |

**TABLE 4 THE DIFFERENT SYMBOLS USED IN THIS ARTICLE AND THEIR DEFINITIONS.**

| SYMBOLS | DEFINITION |
|---|---|
| $x_i$ | the $i$th entry of the vector $\mathbf{x}$ |
| $\mathbf{X}_{ij}$ | the $(i, j)$th entry of the matrix $\mathbf{X}$ |
| $\mathbf{x}_i$ | the $i$th column of the matrix $\mathbf{X}$ |
| $\mathbf{x}_{(j)}$ | the $j$th row of the matrix $\mathbf{X}$ |
| $\|\mathbf{x}\|_0$ | the $l_0$-norm of the vector $\mathbf{x}$—i.e., the number of nonzero entries |
| $\|\mathbf{x}\|_1$ | the $l_1$ norm of the vector $\mathbf{x}$, obtained by $\Sigma_i \lvert x_i \rvert$. |
| $\|\mathbf{x}\|_2$ | the $l_2$ norm of the vector $\mathbf{x}$, obtained by $\sqrt{\Sigma_i x_i^2}$ |
| $\|\mathbf{X}\|_1$ | the $l_1$ norm of the matrix $\mathbf{X}$, obtained by $\Sigma_{i,j}\lvert \mathbf{X}_{ij} \rvert$ |
| $\|\mathbf{X}\|_F$ | the Frobenius norm of the matrix $\mathbf{X}$, obtained by $\sqrt{\Sigma_{i,j}\mathbf{X}_{ij}^2}$ |
| $\hat{\mathbf{X}}$ | the estimate of the matrix $\mathbf{X}$ |
| $\mathrm{tr}(\mathbf{X})$ | the trace of the matrix $\mathbf{X}$ |
| $\|\mathbf{X}\|_{\mathrm{TV}}$ | the TV norm of the matrix $\mathbf{X}$, obtained by $\Sigma_i \mathrm{TV}(\mathbf{x}_{(i)})$ |

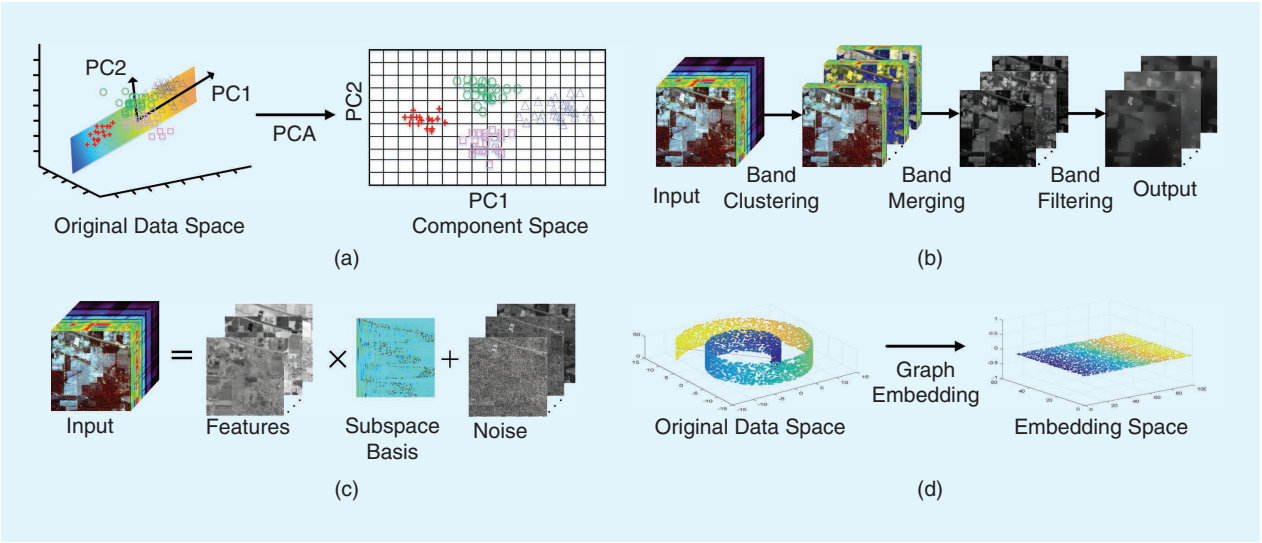


**FIGURE 6.** The four major categories of UFE methods: (a) transform, (b) band-clustering and -merging, (c) low-rank reconstruction, and (d) manifold-learning based.

where

$$\mathrm{TV}(\mathbf{x}) = \left\| \sqrt{(\mathbf{D}_h(\mathbf{x}))^2 + (\mathbf{D}_v(\mathbf{x}))^2} \right\|_1$$

and $\mathbf{D}_v$ and $\mathbf{D}_h$ are the matrix operators, to calculate the first-order vertical and horizontal differences, respectively, of a vectorized image. Recently, sparse and smooth low-rank analysis (SSLRA) was proposed in [43], which models the HSI based on a combination of sparse and smooth features:

$$\mathbf{X} = \mathbf{V}^T(\mathbf{F} + \mathbf{S}) + \mathbf{N}, \tag{8}$$

where $\mathbf{F}$ and $\mathbf{S}$ contain smooth and sparse features, respectively. SSLRA simultaneously extracts the sparse, $\mathbf{S}$, and smooth features, $\mathbf{F}$, by taking into account both sparsity and TV penalties:

$$(\hat{\mathbf{F}}, \hat{\mathbf{S}}, \hat{\mathbf{V}}) = \frac{1}{2}\| \mathbf{X} - \mathbf{V}^T(\mathbf{F} + \mathbf{S}) \|_F^2 + \lambda_1 \| F \|_{\mathrm{TV}} + \lambda_2 \| S \|_1 \quad \text{s.t.} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}. \tag{9}$$

## GRAPH-EMBEDDING AND/OR MANIFOLD-LEARNING TECHNIQUES

Considering the nonlinear characteristic of HSIs, this group of FE techniques aims to capture the data manifold through the local geometric structure of neighboring pixels in the feature space. Figure 6(d) demonstrates the concept of manifold-learning FE techniques applied on the Swiss roll data set. The pink line in the left image shows the Euclidean distance between two data points in 3D space. It is clear that this line is not an effective metric to measure the similarity of the two points selected in the Swiss roll data set.

On the other hand, after unfolding of the data set, which is represented in 2D space in the right image of Figure 6(d), the Euclidean distance between two data points shown by the pink line is a better representation of the similarity of the two points in the data set. The FE techniques categorized in this group are designed to capture such a manifold while representing the data in a lower-dimensional feature space.

Graph-embedding or manifold-learning FE techniques often include three main steps: 1) neighborhood pixel selection, 2) weight selection, and 3) embedding construction. Isometric mapping (ISOMAP) [44], [45] is a global geometric nonlinear FE. ISOMAP searches for geodesic distances between data points and includes three main steps: 1) constructing a neighborhood graph of the data points, 2) computing the shortest path distances between all data points in the neighborhood graph, and 3) creating the lower-dimensional embedding vectors that preserve the path distances in the neighborhood graph.

Locally linear embedding (LLE) [46], Laplacian eigenmaps [47], and locality-preserving projection (LPP) [48] are also geometric nonlinear FE techniques based on graph embedding. LLE constructs the embedding graph in three steps. First, the neighbors for data points are selected using the $K$ nearest neighbors. Second, the weights $\mathbf{A}_{i,j}$ that linearly reconstruct the data points are computed using their neighbors by minimizing the following constrained least squares:

$$\min_{\mathbf{A}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j \in \phi_i} \mathbf{A}_{ij}\mathbf{x}_j \right\|_2^2 \quad \text{s.t.} \quad \sum_{j \in \phi_i} \mathbf{A}_{ij} = \mathbf{1}, \tag{10}$$

where $\phi_i(\mathbf{x}_i)$ contains the neighborhood pixels selected for $\mathbf{x}_i$. We should note that the constrained weights estimated from (10) for every data point are invariant to rotations, rescalings, and translations of that data point and its neighbors; therefore, they characterize the intrinsic geometric

properties of each neighborhood. Third, the lower-dimensional embedding vectors $\mathbf{y}$ are constructed by minimizing

$$\min_{\mathbf{z}} \sum_{i=1}^{n} \left\| \mathbf{z}_i - \sum_{j \in \phi_i} \mathbf{A}_{ij} \mathbf{z}_j \right\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{n} \mathbf{z}_i = \mathbf{0}, \quad \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^T = \mathbf{I}. \tag{11}$$

We should note that the reconstruction weights $\mathbf{A}_{ij}$ are fixed in minimization (11), and, therefore, the intrinsic geometric properties of the data with dimension $p$ are invariant to such a transformation into a lower-dimension $d$.

In [49], a general framework for graph embedding is given by

$$\min_{\mathbf{z}} \sum_{i,j \in \phi_i} \| \mathbf{z}_i - \mathbf{z}_j \|_2^2 \mathbf{W}_{ij} \quad \text{s.t.} \quad \mathbf{ZBZ}^{\mathrm{T}} = \mathbf{I} \tag{12}$$

or, equivalently,

$$\min_{\mathbf{z}} \operatorname{tr}(\mathbf{Z}(\mathbf{D} - \mathbf{W})\mathbf{Z}^T) = \min_{\mathbf{z}} \operatorname{tr}(\mathbf{ZLZ}^{\mathrm{T}}) \quad \text{s.t.} \quad \mathbf{ZBZ}^{\mathrm{T}} = \mathbf{I}, \tag{13}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ denotes the Laplacian matrix of the undirected weighted graph $G = \{\mathbf{X}, \mathbf{W}\}$ (where $\mathbf{X}$ is the vertex set and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the similarity matrix) and $\mathbf{D}$ is a diagonal matrix where its entries are given by

$$\mathbf{D}_{ii} = \sum_{j \neq i} \mathbf{W}_{ij}, \forall i. \tag{14}$$

The diagonal matrix $\mathbf{B}$ is for the scale normalization and might also be the Laplacian matrix of a penalty graph, such as $G^p = \{\mathbf{X}, \mathbf{W}^p\}$. We should note that the vertices of $G^p$ and $G$ (i.e., $\mathbf{X}$) are the same, while the similarity matrix ($\mathbf{W}^p$) corresponds to the similarity characteristics suppressed in the lower-dimensional feature space ($\mathbf{B} = \mathbf{L}^p = \mathbf{D}^p - \mathbf{W}^p$; see [49]). LLE can be reformulated using the graph embedding mentioned earlier with similarity matrix $\mathbf{W}_{ij} = \mathbf{A}_{ij} + \mathbf{A}_{ij}^T - \mathbf{A}_{ij}^T \mathbf{A}_{ij}$ if $j \in \phi_i$; otherwise, $\mathbf{W}_{ij} = 0$ and $\mathbf{B} = \mathbf{I}$ [49]. ISOMAP, LE, and LPP can also be formulated using graph embedding [49]. From the viewpoint of graph embedding, the main difference between these FE techniques is the selection of the matrices $\mathbf{W}$ and $\mathbf{B}$. For instance, LE and LPP use the Gaussian function with the standard deviation $\sigma$ to choose the similarity matrix as

$$\mathbf{W}_{ij} = \begin{cases} \exp \dfrac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}, & \forall i, j \in \phi_i(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

We should note that the techniques categorized in this group are assumed to be SFE methods when they are applied only on the training samples. This is common in the case of HSI due to the large volume of the image, which makes the algorithm computationally very expensive. In the following section, we discuss how the ground reference (training samples) can be used to construct the edge matrix, $\mathbf{W}$; therefore, those techniques are considered SFE.

### *SFE TECHNIQUES*

Unlike UFE techniques that rely on modeling various prior assumptions of hyperspectral data, supervised methods are capable of extracting class-separable features more effectively, owing to the use of label information. Over the past few decades, some seminal models have been widely developed and applied to perform SFE on HSIs; these can be roughly categorized into two streams: subspace-learning (SL)-based and band-selection (BS)-based approaches.

Different from handcrafted features [50], SL-based approaches learn to extract the low-dimensional representation from the data by formulating different supervised rules in view of label information. There are some typical methods in SL, including LDA [51], matrix discriminant analysis (MDA) [52], decision boundary FE [53], and so on. The latter BS-based methods, which aim at screening out the representative and informative spectral bands, are unfolded with MI-based BS [54], rough set, and fuzzy C-means [55], to name a few. To further enhance class separability, many extended methods have been successfully proposed in recent years: subspace LDA (SLDA) [56], regularized LDA [57], local Fisher's discriminant analysis (LFDA) [58], feature space discriminant analysis (FSDA) [59], rough-set-based BS [60], and FE with local spatial modeling [61].

Because of the powerful learning ability of SL methods compared to that of BS-based strategies, we focus on reviewing the SL-related FE techniques, in which two main streams—discriminant analysis FE (DAFE) and regression-induced representation learning (RIRL)—are emphatically investigated and compared by clarifying their similarities and differences as well as pros and cons, as briefly illustrated in Figure 7.

#### DAFE

Generally speaking, DAFE seeks to find an optimal projection or transformation matrix $\mathbf{P} \in \mathbb{R}^{p \times d}$ ($d$ is the dimension of the subspace to be estimated) by optimizing certain class-relevant separation criteria associated with the label information. In this process, the estimated subspace $\mathbf{Z} \in \mathbb{R}^{d \times n}$, which consists of a series of vector $\mathbf{z}_i$, can be obtained by projecting the samples $\mathbf{X}_m = \{\mathbf{x}_i\}_{i=1}^m \in \mathbb{R}^{p \times m}$ onto a decision boundary, which can be generally expressed as $\mathbf{Z} = \mathbf{P}^T \mathbf{X}$. Each vector $\mathbf{z}_i$ in $\mathbf{Z}$ can be collected by $\mathbf{P}^{\mathrm{T}} \mathbf{x}_i$. Depending on the different types of label embedding, DAFE can be subdivided into LDA and its variants, graph-embedding-based discriminant analysis (GDA) and its extensions, and kernelized discriminant analysis (KDA).

#### *LDA AND ITS VARIANTS*

Traditional LDA linearly transforms the original data into a discriminative subspace by maximizing the Fisher's ratio in the form of the generalized Rayleigh quotient, that is, minimizing the intraclass scatter and maximizing interclass scatter simultaneously. Given a pairwise training set $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_i, \mathbf{y}_i), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$, the objective function of

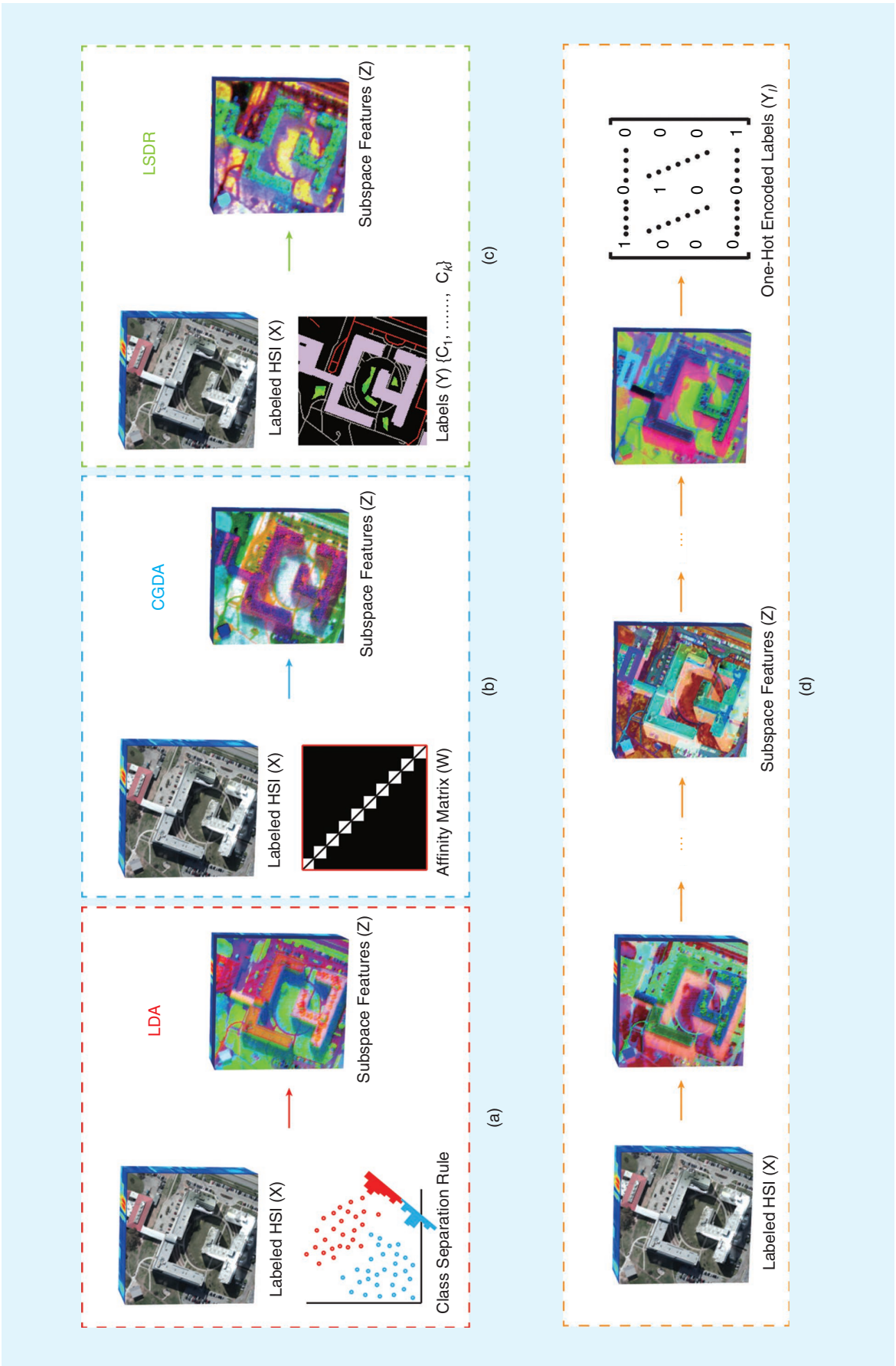


**FIGURE 7.** SFE with four different categories: (a) class-separation-based discriminant analysis, (b) collaborative graph-embedding-based discriminant analysis (CGDA), (c) regression-based representation learning, and (d) joint and progressive learning strategy (Jplay). The obvious differences lie in the use form of label information and learning strategies, i.e., LDA: Fisher's rule; CGDA: affinity matrix; least-squares dimension reduction:(LSDR) labels; JPlay: joint use of affinity matrix and one-hot encoded labels.

multiclass LDA to estimate the linear projection matrix $\mathbf{P}$ can be written as follows:

$$\max_{\mathbf{P}} \frac{\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_b\mathbf{P})}{\mathrm{tr}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_w\mathbf{P})}, \tag{16}$$

where $\mathbf{S}_w$ and $\mathbf{S}_b$ are defined as the within-class and between-class scatter matrices, respectively. With the constraint of $\mathbf{P}^{\mathrm{T}}\mathbf{S}_w\mathbf{P}=\mathbf{I}$, the optimization problem in (16) can be equivalently converted to one of $\mathbf{S}_b\mathbf{P}=\lambda\mathbf{S}_w\mathbf{P}$ by introducing the Lagrange multiplier $\lambda$. The close-form solution to the simplified optimization problem can be deduced by generalized eigenvalues decomposition (GED).

Due to the sensitivity to complex, high-dimensional noises caused by environmental and instrumental factors and the availability of labeled samples, the original LDA inevitably suffers from an ill-posed statistical degradation, especially in the case of small-scale samples. The degraded reasons mainly lie in the singularity of the two scatter metrics ($\mathbf{S}_w$ and $\mathbf{S}_b$), thereby easily leading to the overfitting problem. To improve stability and generalization, the regularized LDA was proposed by adding an $l_2$-norm constraint on $\mathbf{S}_w$, parameterized by $\gamma$ as $\mathbf{S}_w^{\mathrm{reg}}=\mathbf{S}_w+\gamma\mathbf{I}$. By replacing $\mathbf{S}_w$ in (16) with the regularized $\mathbf{S}_w^{\mathrm{reg}}$, the solution in the regularized LDA can be still obtained by the GED solver.

Considering the local neighborhood relations between samples in the process of model learning, LFDA breaks through the bottleneck of those LDA-based methods by assuming that the data are distributed in the nonlinear manifolds rather than a homogeneous Gaussian space. For this purpose, LFDA is capable of effectively excavating the locally underlying structure of the data that lie in the real world. Essentially, LFDA can be regarded as a weighted LDA by locally weighing $\mathbf{S}_w$ and $\mathbf{S}_b$ matrices. Therefore, the two modified scatter matrices, denoted as $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$, can be formulated as

$$\begin{aligned}\tilde{\mathbf{S}}_w &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{W}_{ij}^{w}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^{\mathrm{T}},\\ \tilde{\mathbf{S}}_b &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{W}_{ij}^{b}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^{\mathrm{T}},\end{aligned} \tag{17}$$

where the two weights ($\mathbf{W}^w$ and $\mathbf{W}^b$) denote the samplewise similarities. There are several commonly used strategies for calculating such a similarity matrix symbolized by $\mathbf{W}$. A simple yet effective one is given by $\mathbf{W}_{ij}=1$, if $\mathbf{x}_j\in\phi_k(\mathbf{x}_i)$, where $\phi_k(\mathbf{x}_i)$ represents the k-nearest neighbor of $\mathbf{x}_i$; otherwise, $\mathbf{W}_{ij}=0$. Another commonly used technique was constructed based on the radial basis function with a standard derivation of $\sigma$, as defined in (15). Refer to [62]–[64], which might be useful for those who are interested in more types of $\mathbf{W}$.

Similar to SLDA, which first projects the original data into a subspace and then LDA is performed in the transformed subspace, FSDA starts with maximizing the between-spectral scatter matrix ($\mathbf{S}_f$) to enhance the differences along the spectral dimension; similarly, LDA is further used to extract the representations of class separability from the feature domain. In the first step, let $\mu_{i,j}$ be the average value of the $j$th class and the $i$th spectral band. Then, we have the definition of $\mathbf{S}_f$ as follows:

$$\mathbf{S}_f=\frac{1}{2}\sum_{i=1}^{p}(\mathbf{h}_i-\bar{\mathbf{h}})(\mathbf{h}_i-\bar{\mathbf{h}})^{\mathrm{T}}, \tag{18}$$

where $\mathbf{h}_i=[\mu_{i,1},\mu_{i,2},\ldots,\mu_{i,k}]$ is the spectral representation in the feature space and $\bar{\mathbf{h}}=(1/p)\sum_{i=1}^{p}\mathbf{h}_i$. The primary transformation ($\mathbf{P}_f$) that aims at improving spectral discriminant can be estimated by maximizing the trace term of $\mathbf{S}_f$ as

$$\max_{\mathbf{P}_f}\mathrm{tr}(\mathbf{P}_f^{\mathrm{T}}\mathbf{S}_f\mathbf{P}_f). \tag{19}$$

Using the obtained $\mathbf{P}_f$, the latent representation in the feature space $\mathbf{g}_i=\mathbf{P}_f^{\mathrm{T}}\mathbf{h}_i$, $i=1,2,\ldots,p$ can be further fed into the next step, LDA.

### GDA AND ITS EXTENSIONS

Before revisiting the GDA methods, we first introduce and formulate the general graph embedding (GGE) framework presented in [49] with (12). Obviously, the extracted features $\mathbf{Z}$ in the GGE framework are determined by the construction of $\mathbf{W}$ to a great extent. Thus, we highlight several types of representative affinity matrices corresponding to the different graph-embedding approaches, i.e., LDA, LE [47] and its linearized LPP [48], LLE [46], sparse GDA (SGDA) [65], and collaborative GDA (CGDA) [66]. Figure 8 visualizes the

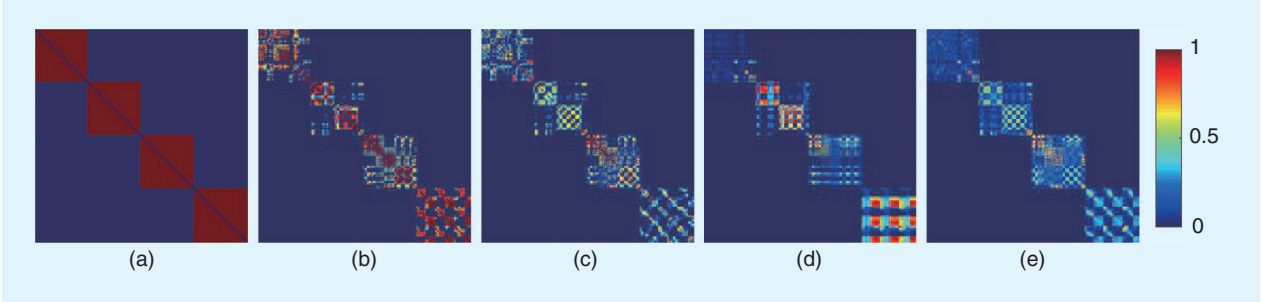


**FIGURE 8.** A four-class showcase for affinity matrices ($\mathbf{W}$) with respect to five different approaches, where the connectivity (or edge) of $\mathbf{W}$ is computed within each class: (a) LDA, (b) LPP, (c) LLE, (d) SGDA, and (e) CGDA.

affinity matrices given by five different strategies in a four-class case selected from the Houston 2013 data set.

### LDA-LIKE AFFINITY MATRIX

In essence, LDA is vested in a special case of the GGE framework with $\mathbf{D}^{(\mathrm{LDA})}=\mathbf{I}$, whose affinity matrix can be represented as

$$\mathbf{W}_{ij}^{(\mathrm{LDA})}=\begin{cases}1/N_k, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j\in C_k;\\ 0, & \text{otherwise,}\end{cases} \tag{20}$$

where $N_k$ is the number of samples belonging to $k$th class.

### LPP- OR LE-BASED AFFINITY MATRIX

One is to be constructed in kernel space with a higher dimension via similarity measurement, i.e., extensively using (15).

### LLE-BASED AFFINITY MATRIX

Different from the handcrafted graph, LLE reconstructs each given sample with its $k$-nearest neighbors by exploiting linear regression techniques [67], [68]. As a result, the reconstruction coefficients ($\mathbf{A}$) can be obtained by solving the optimization problem of (10). With the known $\mathbf{A}$, it is straightforward to derive the needful affinity matrix, denoted as $\mathrm{W}^{(\mathrm{LLE})}$,

$$\mathbf{W}_{ij}^{(\mathrm{LLE})}=\begin{cases}\mathbf{A}_{ij}+\mathbf{A}_{ij}^{\mathrm{T}}-\mathbf{A}_{ij}\mathbf{A}_{ij}^{\mathrm{T}}, & \text{if } \mathbf{x}_j\in\phi_k(\mathbf{x}_i);\\ 0, & \text{otherwise,}\end{cases} \tag{21}$$

thereby inducing the Laplacian matrix as $\mathbf{L}^{(\mathrm{LLE})}=\mathbf{D}^{(\mathrm{LLE})}-\mathbf{W}^{(\mathrm{LLE})}=(\mathbf{I}-\mathbf{A})^{\mathrm{T}}(\mathbf{I}-\mathbf{A})$.

### SGDA- AND CGDA-GUIDED AFFINITY MATRIX

Similar to LLE, the affinity matrix can be estimated using data-driven representation learning, i.e., sparse and collaborative representations [69]–[71]. Accordingly, the two learning strategies can be equivalent to respectively solving the constrained $l_1$-norm optimization problem,

$$\min_{\mathbf{W}}\|\mathbf{W}\|_1 \quad \text{s.t.} \quad \|\mathbf{X}_m\mathbf{W}-\mathbf{X}_m\|_F^2\le\epsilon, \tag{22}$$

and the $l_2$-norm optimization problem,

$$\min_{\mathbf{W}}\|\mathbf{W}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{X}_m\mathbf{W}-\mathbf{X}_m\|_F^2\le\epsilon. \tag{23}$$

These affinity matrices can be unified to the GGE framework of (12).

In addition to SGDA and CGDA (the two baselines), Huang et al. [72] learned a set of sparse coefficients on manifolds and then preserved the sparse manifold structure in the embedded space. In [73], Xue et al. extended the existing SGDA to the spatial–spectral graph embedding to address issues of spatial variability and spectral multimodality. With the embedding of the intrinsic geometric structure of the data, a Laplacian regularizer CGDA [74] was developed to further improve the graph's confidence. Li et al. [75] simultaneously integrated sparsity and low rankness into the graph to capture a more robust structure of the data locally and globally. Furthermore, Pan et al. [76] improved the work by Li et al. [75] by unfolding the HSI data with the form of a tensor.

### KDA

In reality, the HSI usually exhibits a highly nonlinear data distribution, which may result in difficulties in effectively identifying the materials. The solution to this issue makes use of a so-called kernel trick [77] that can map the data of the input space into a new Hilbert space with a higher feature dimension. In the kernel-induced space, the complex nonlinearity of the HSI can be well analyzed in a linearized system. Comparatively, the input to KDA is an inner product of original data pairs, defined as $k(\mathbf{x}_i,\mathbf{x}_j)$, which can be given by (15). By introducing the kernel Gram matrix $\mathbf{K}$ with $\mathbf{K}_{i,j}=\mathbf{\Phi}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}_j)=k(\mathbf{x}_i,\mathbf{x}_j)$, most of the previous LDA-based methods can be simply extended to the corresponding kernelized versions; i.e., kernelized LDA (KLDA) and kernelized LFDA (KLFDA) can calculate their projections $\mathbf{P}$ by solving a GED problem of

$$\mathbf{KLKP}=\lambda(\mathbf{KBK}+\gamma\mathbf{I})\mathbf{P}. \tag{24}$$

Note that $\mathbf{B}=\mathbf{I}$ in KLDA, whereas $\mathbf{L}=\mathbf{L}_w$ and $\mathbf{B}=\mathbf{L}_b$ are computed by $\mathbf{D}_w-\mathbf{W}_w$ and $\mathbf{D}_b-\mathbf{W}_b$ in the kernel space, respectively, for KLFDA. Furthermore, for kernelized SGDA (KSGDA) and kernelized GCDA (KCGDA), the main difference lies in the computation of the adjacency matrix, which can be performed in the kernel space by solving the general kernel coding problem as follows:

$$\min_{\mathbf{W}}\mathbf{\Omega}(\mathbf{W}) \quad \text{s.t.} \quad \|\mathbf{\Phi}(\mathbf{X}_m)\mathbf{W}-\mathbf{\Phi}(\mathbf{X}_m)\|_F^2\le\epsilon, \tag{25}$$

where $\mathbf{\Omega}(\mathbf{W})$ can be selected to be either the sparsity-prompting term $\|\mathbf{W}\|_1$ of KSGDA or the dense (or collaborative) term $\|\mathbf{W}\|_F^2$ of KCGDA. In [78] and [74], the solutions in (25) were theoretically guaranteed in the same way by solving (22) and (23) using the alternating-direction method of multiplier (ADMM) [79] and least-square regression with Tikhonov regularization [80], respectively.

### RIRL

RIRL provides a new insight from the regression point of view to model the FE behavior by bridging the training samples with the corresponding labels rather than indirectly using the label information in the form of a graph or affinity matrix in DAFE-based methods.

### LEAST-SQUARES DIMENSION REDUCTION

We begin with sliced inverse regression [81], which is a landmark in SFE techniques. It assumes that the pairwise data $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^m$ are conditionally independent on the to-be-estimated subspace features $\{\mathbf{z}_i\}_{i=1}^m$, formulated as $(\mathbf{X}\perp\mathbf{Y})|\mathbf{Z}$. Following this rule, the least-squares dimension reduction (LSDR) proposed by Suzuki and Sugiyama [82]

attempts to find a maximizer of the squared-loss MI (SMI) to satisfy the previously mentioned independence assumption. The projections $\mathbf{P}$ for LSDR can be searched by optimizing the following maximization problem:

$$\max_{\mathbf{P}} \mathrm{SMI}(\mathbf{Z}, \mathbf{Y}) \quad \text{s.t.} \quad \mathbf{P}\mathbf{P}^{\mathrm{T}} = \mathbf{I}. \tag{26}$$

And the SMI to measure a statistical dependence between two discrete variables is defined as

$$\mathrm{SMI}(\mathbf{Z}, \mathbf{Y}) = \sum_{\mathbf{z}\in\mathbf{Z}} \sum_{\mathbf{y}\in\mathbf{Y}} p(\mathbf{z})p(\mathbf{y}) \left( \frac{p(\mathbf{z}, \mathbf{y})}{p(\mathbf{z})p(\mathbf{y})} - 1 \right)^2, \tag{27}$$

where $p(\cdot)$ is the probability distribution function.

### *LEAST-SQUARES QUADRATIC MI*

Limited by the sensitivity of MI to outliers, Sainui and Sugiyama [83] designed a more robust least-squares quadratic MI (LSQMI) on the basis of a QMI criterion; hence, let us define the QMI as

$$\mathrm{QMI}(\mathbf{Z}, \mathbf{Y}) = \sum_{\mathbf{z}\in\mathbf{Z}} \sum_{\mathbf{y}\in\mathbf{Y}} (p(\mathbf{z}, \mathbf{y}) - p(\mathbf{z})p(\mathbf{y}))^2. \tag{28}$$

Similarly, we solve (26)-like optimization problem by replacing SMI with QMI.

### *LSQMI DERIVATIVE*

Due to the difficulty in accurately computing the derivative of the QMI estimator, LSQMI was further extended to a computationally effective LSQMI derivative by estimating the derivative of QMI instead of QMI itself [84]. In that article, Tangkaratt et al. [84] demonstrated a more accurate and efficient derivative computation of QMI.

### *JOINT AND PROGRESSIVE LEARNING STRATEGY*

Another MI-free estimation group is latent SL (LSL). One representative LSL performs FE and classification simultaneously in a joint-learning (JL) fashion [85]. With an expected output $\mathbf{\Theta}\mathbf{X}_m$, the process can be modeled as

$$\min_{\mathbf{P}_k, \mathbf{\Theta}} \| \mathbf{Y}_l - \mathbf{P}_k \mathbf{\Theta} \mathbf{X}_m \|_F^2 + \frac{\alpha}{2} \| \mathbf{P}_k \|_F^2 \quad \text{s.t.} \quad \mathbf{\Theta}\mathbf{\Theta}^{\mathrm{T}} = \mathbf{I}, \tag{29}$$

where $\mathbf{Y}_l \in \mathbb{R}^{k\times m}$ and $\mathbf{\Theta} \in \mathbb{R}^{d\times p}$ are defined as the one-hot encoded label matrices and the latent subspace projections, respectively. $\mathbf{P}_k \in \mathbb{R}^{k\times d}$ denotes the regression matrix that connects the learned subspace and the label information. $\mathbf{Y}_l$ can be formulated as

$$\mathbf{Y}_l = \begin{bmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ & & \dots & \dots & \dots & \\ 0 & 0 & \dots & 1 & \dots & 0 \\ & & \dots & \dots & \dots & \\ 0 & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ \dots \\ j \\ \dots \\ k \end{matrix}. \tag{30}$$

In [85], the model's solution was proven to be a closed form. Moreover, in [86], Hong et al. explored an LDA-like graph as a regularizer to learn a spectrally discriminative feature representation; thus, (29) becomes

$$\min_{\mathbf{P}_k, \mathbf{\Theta}} \| \mathbf{Y}_l - \mathbf{P}_k \mathbf{X}_m \|_F^2 + \frac{\alpha}{2} \| \mathbf{P}_k \|_F^2 + \frac{\beta}{2} \mathrm{tr}(\mathbf{\Theta}\mathbf{X}_m \mathbf{L} \mathbf{X}_m^{\mathrm{T}} \mathbf{\Theta}^{\mathrm{T}}) \quad \text{s.t.} \ \mathbf{\Theta}\mathbf{\Theta}^{\mathrm{T}} = \mathbf{I}. \tag{31}$$

Beyond the JL-based models, Hong et al. [87] established a novel multilayered regression framework by following a joint and progressive learning strategy (JPlay). With the layerwise autoreconstruction mechanism effective against the spectral variabilities caused by complex noises and atmospheric effects, the linearized JPlay breaks through the performance bottleneck of traditional linear methods. More specifically, we have the resulting model

$$\begin{aligned} \min_{\mathbf{P}_k, \{\mathbf{\Theta}_l\}_{l=1}^q} & \| \mathbf{Y}_l - \mathbf{P}_k \mathbf{\Theta}_q \dots \mathbf{\Theta}_1 \mathbf{X}_m \|_F^2 + \frac{\alpha}{2} \| \mathbf{P}_k \|_F^2 \\ & + \frac{\beta}{2} \sum_{l=1}^{q} \mathrm{tr}(\mathbf{\Theta}_l \mathbf{X}_{l-1} \mathbf{L} \mathbf{X}_{l-1}^T \mathbf{\Theta}_l^T) \\ & + \frac{\gamma}{2} \sum_{l=1}^{q} \| \mathbf{X}_{l-1} - \mathbf{\Theta}_l^T \mathbf{\Theta}_l \mathbf{X}_{l-1} \|_F^2, \\ \text{s.t.} \ & \mathbf{X}_l = \mathbf{\Theta}_l \mathbf{X}_{l-1}, \mathbf{X}_l \geq 0, \| \mathbf{x}_i \|_2 \leq 1, \end{aligned} \tag{32}$$

where the soft constraint $\| \mathbf{x}_i \|_2 \leq 1$ can be used to relax the orthogonality. It is worth noting that such a JL-based strategy can clearly tell the model which features are positive to the classification task, owing to the joint strategy of FE and classification.

## DEEP FE TECHNIQUES

Shallow FE techniques often require careful engineering and domain knowledge of experts, which limits their applications. In contrast, DL techniques aim at automatically learning high-level features from raw data in a hierarchical fashion. These features are more discriminative, abstract, and robust than those in shallow methods. Due to their powerful feature representation ability, DL techniques have been widely used to extract features from HSIs in recent years [88], [89]. Among various DL models, autoencoders (AEs), convolutional neural networks (CNNs), and recurrent NNs (RNNs), shown in Figure 9, are the most popular. In this section, we present these models and their applications to hyperspectral FE.

### *AES*

As demonstrated in Figure 9, AE mainly comprises two modules: encoder and decoder. Encoder maps the input vector $\mathbf{x}$ into a hidden space $\mathbf{h}$, whereas decoder aims at getting a reconstruction result $\hat{\mathbf{x}}$ of the original input from $\mathbf{h}$. These processes can be formulated as

$$\begin{aligned} \mathbf{h} &= f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1), \\ \hat{\mathbf{x}} &= f(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2), \end{aligned} \tag{33}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ denote the weights connecting the input layer to the hidden layer and the hidden layer to the output

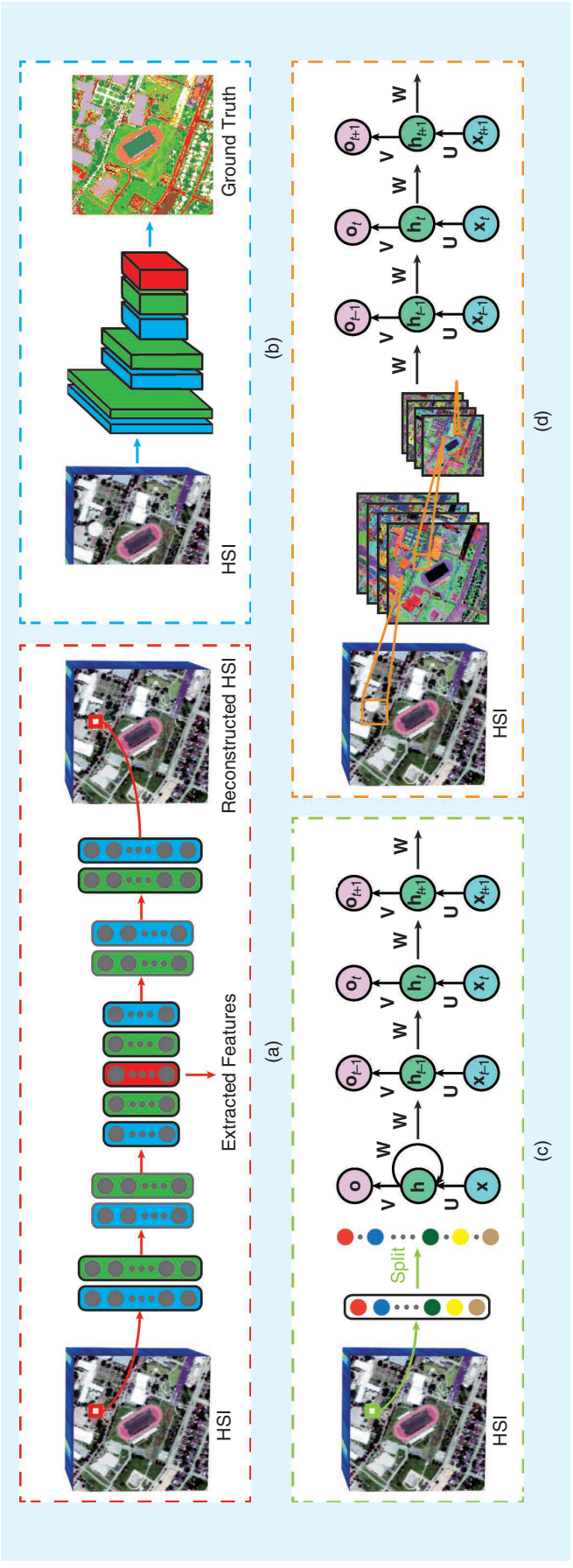


**FIGURE 9.** The four major categories of DL models: the (a) AE-based, (b) CNN-based, (c) RNN-based, and (d) integrated DL models.

layer, respectively; $\mathbf{b}_1$ and $\mathbf{b}_2$ represent the biases of the hidden units and output units, respectively; and f is a nonlinear activation function. The training of an AE is to minimize the residual between $\mathbf{x}$ and $\hat{\mathbf{x}}$. Once trained, the decoder is deleted, and the hidden layer $\mathbf{h}$ is considered as a feature representation of $\mathbf{x}$. To extract deep features, several AEs are often stacked together, generating a stacked AE (SAE) model. For SAE, the hidden layer in the preceding AE is used as the input of the subsequent AE.

SAE is, perhaps, the earliest deep model used to extract features of HSIs [90]. One typical benefit of SAEs is that each AE inside the network can be pretrained using both labeled and unlabeled samples, thus providing better initial values for network parameters compared to random initialization. After layerwise pretraining, the fine-tuning of only a few layers can acquire satisfactory discriminant features. This training method is capable of alleviating the overfitting problem when there exist only small numbers of training samples in HSIs. In [91], the spectral information for each pixel was considered as a vector and fed into an SAE model to extract deep features. These features extracted by SAEs can also be generalized from one image to another image, which was validated in [92] and [93].

To extract the spatial features of each pixel, one often needs to select a local patch or cube centered at the pixel and then input it into an FE model. Since the inputs of SAEs are vectors, it is difficult to directly process patches or cubes. In [91] and [92], the local cubes from the first principal components of HSIs were initially reshaped into vectors and then fed into SAEs to extract spatial features. In [94] and [95], Gabor features and extended morphological attribute profiles (i.e., the joint use of shallow and deep FE methods) were used as the inputs of SAEs, making it easier for the network to extract high-level spatial features. After the extraction of spectral and spatial features, these features can be easily concatenated together to generate a spectral–spatial joint feature [91], [92]. Compared to the concatenation method, Kang et al. [94] and Deng et al. [95] proposed using another SAE to fuse the spectral and spatial features, which may further enhance the discriminative ability of spectral–spatial features.

Similar to traditional FE methods, one can also embed some prior or expected information into SAEs. Based on the assumption that neighboring samples in the input space should have similar hidden representations, graph regularization was added to SAE to preserve this property [96], [97]. In [98], Zhou et al. imposed a local regularization via FDA on hidden layers to make the extracted features of samples from the same category close to each other and those from different categories as far apart as

possible, thus improving the discriminative ability of the SAE. Meanwhile, they also added a diversity regularization term to make the SAE extract compact features.

### CNNs

CNNs are the most popularly adopted deep model for hyperspectral FE. As shown in Figure 9, the basic components of a CNN model include convolutional layers, pooling layers, and fully connected layers. The convolutional layers are used to extract features with convolutional kernels (filters), which can be formulated as

$$\mathbf{X}^l = f(\mathbf{X}^{l-1} * \mathbf{W}^l + b^l), \tag{34}$$

where $\mathbf{X}^l$ is the $l$th feature maps; $\mathbf{W}^l$ and $\mathbf{b}^l$ denote the filters and biases of the $l$th layer, respectively; and $*$ represents the convolutional operation. After the convolutional layer, the pooling layer is often adopted to reduce the size of the generated feature maps and produce more robust features. On the top of a CNN model, there often exist some fully connected layers, aiming at learning high-level features and outputting the final results of the network.

For HSIs, CNNs can be used to extract spectral features [99] or spatial features [100]–[102], depending on the inputs of networks. In [99], Hu et al. designed a 1D CNN model to extract spectral features of each pixel. Compared to traditional fully connected networks, CNNs have weight-sharing and local-connection characteristics, making their training processes more efficient and effective. In [100], 2D CNN was explored to extract spatial features from a local cube. Different from SAEs, CNNs do not need to reshape the cube into a vector, thus preserving as much spatial information as possible. However, to make full use of the representation ability of CNNs, two important issues need to be considered. The first issue is the small number of training samples but high-dimensional spectral information, which will easily lead to the overfitting problem. The second issue is the extraction of spectral–spatial joint features, which can improve the classification performance in comparison with using the spectral or spatial feature only.

For the first issue, many commonly used strategies in the field of natural image classification, such as dropout and weight decay, can be adopted. In addition, many promising methods have been proposed in the past few years. These methods can be divided into four different classes.

The first class of methods is dimensionality reduction. In [100], [101], and [103], PCA was employed to extract the first principal components of HSIs as inputs of CNNs, thus simplifying the network structures. Similarly, a similarity-based BS method was used in [104]. However, these dimensionality-reduction methods are independent from the following CNNs, which may lose some useful information. Different from them, Ghamisi et al. [105] proposed a novel method to adaptively select the most informative bands suitable for the CNN model.

The second class of methods is data augmentation. In [101], two methods were proposed to generate virtual samples. One is to multiply a random factor and add a random noise to training samples, while the other is to combine two given samples from the same class with proper ratios. In [106], a data-augmentation method based on distance density was proposed. Recently, Kong et al. [107] proposed a random zero-setting method to generate new samples.

The third class of methods is transfer learning. In [108] and [109], the authors found that CNNs trained by one hyperspectral data set can be transferred to another data set acquired by the same sensor and that fine-tuning only a few top layers achieves satisfying results. More interestingly, the works in [110]–[112] indicated that CNNs pretrained by natural images can be directly applied to extract spatial features of HSIs.

The fourth class of methods is semisupervised or even unsupervised learning. For example, Wu and Prasad [113] attempted to use a clustering model to obtain pseudo-labels of unlabeled samples and then combine the training samples and unlabeled samples (with their pseudo-labels) together to train their network.

In terms of the second issue, one popularly used method is feeding a local cube, directly cropped from the original HSI, into a CNN with 3D convolution kernels for processing the spectral and spatial information simultaneously. The number of channels in the 3D convolutional kernel is smaller than or equal to that of its input layer. However, the former dramatically increases the computational complexity due to the simultaneous convolution operators in both the spectral and spatial domains, whereas the latter heavily increases the number of parameters to optimize. Another candidate method is to decouple the task of spectral–spatial FE into two parts: spectral FE and spatial FE.

In [108] and [114], a parallel structure was employed to extract spectral–spatial features. Specifically, 1D and 2D CNNs were designed to extract spectral features and spatial features, respectively; these two features were then concatenated together and fused via a few fully connected layers. Since 2D CNN focuses on extracting spatial features, some redundant spectral information can be preprocessed to reduce the computational complexity. In [115], a serial structure was also used to extract spectral–spatial features. It first applied several $1 \times 1$ convolutions to extract spectral features and then fed the extracted features into several 3D convolutions to extract spatial features.

### RNNs

RNNs have been popularly employed to sequential data analysis, such as machine translation and speech recognition. Different from the feedforward NN, RNN takes advantage of a recurrent edge to connect the neuron to itself across time. Therefore, it is able to model the probability distribution of sequence data. To make this subsection easier to follow, we first provide a brief and general discussion

on RNN. Then, we briefly describe how to use RNN specifically for the classification of HSIs.

Figure 9 shows an example of RNN. Given a sequence $\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t$, $t \in \{1, 2, \ldots, T\}$ generally denotes the information at time $t$, the output of the hidden layer at the $t$th time step is

$$\mathbf{h}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1} + \mathbf{b}_h), \tag{35}$$

where $\mathbf{U}$ and $\mathbf{W}$ represent weight matrices from the current input layer to the hidden layer and the preceding hidden layer to the current hidden layer, respectively; $\mathbf{h}_{t-1}$ is the output of the hidden layer at the preceding time; and $\mathbf{b}_h$ is a bias vector. According to this equation, it can be observed that the contextual relationships in the time domain are constructed via a recurrent connection. Ideally, $\mathbf{h}_T$ will capture most of the information and can be considered the final feature of the sequence data. In terms of classification tasks, one often inputs $\mathbf{h}_t$ into an output layer $\mathbf{o}_t$, which can be described as

$$\mathbf{o}_t = f(\mathbf{V}\mathbf{h}_t + \mathbf{b}_o), \tag{36}$$

where $\mathbf{V}$ is the weight matrix from the hidden layer to the output layer and $\mathbf{b}_o$ is a bias vector.

In recent years, RNNs have attracted more and more attention in the field of HSI FE. To make full use of RNNs, one must first ask the following question: How is the sequence to be constructed? An intuitive method is to regard the whole spectral bands as a sequence [116], [117]. For each pixel, its spectral values are fed into RNNs from the first band to the last band, and the output of the hidden layer at the last band is the extracted spectral feature. Different from the traditional sequences in speech-recognition or machine-translation tasks, the succeeding bands do not depend on the preceding ones. Thus, Liu et al. [116] also fed the spectral sequence from the last band to the first band to construct a bidirectional RNN model.

Another method is to use a local patch or cube to construct the sequence [117]–[119]. For example, Zhou et al. [117] regarded the rows of each local patch, cropped from the first principal component of HSIs, as a sequence and fed them into an RNN one by one to extract spatial features; Zhang et al. [118] adopted each pixel and its neighboring pixels in the cube to form a sequence. These pixels were first sorted according to their similarities to the center pixel and then fed into the RNN sequentially to extract locally spatial features.

In real applications, the constructed sequence may be very long. In the widely used Indian Pines data, the length of the sequence is 200 (the number of spectral bands) if we use the first method mentioned earlier to construct the sequence. This sequence increases the training difficulty because the gradients tend to either vanish or explode. To deal with this issue, long short-term memory (LSTM) was employed as a more sophisticated recurrent unit [116], [117], [120], [121]. The core components of LSTM are three gates: input, forget, and output gates. These gates together control the flow of information in the network.

Similarly, the gated recurrent unit (GRU), which has only two gates (i.e., an update gate and a reset gate), was also employed. Compared to LSTM units, GRUs have fewer parameters, which may be more suitable for HSI FE since it usually has a limited number of training samples. Another candidate scheme to address the issue is to divide the long-term sequence into shorter sequences [121], [122]. For example, in [122], Hang et al. proposed grouping the adjacent bands of HSIs into subsequences and then using RNNs to extract features from them. Since nonadjacent bands have some complementarity, they also used another RNN to fuse the extracted features.

### *INTEGRATED NETWORKS*

In general, AEs and RNNs are good at processing vectorized inputs, thus achieving promising results in terms of spectral FE. However, both of them need to reshape the input patches or cubes into vectors during spatial FE, which may destroy some spatial information. In contrast, CNNs are able to directly deal with image patches and cubes, resulting in more powerful spatial features than AEs and RNNs. It is natural to wonder whether we can integrate these networks together to make full use of their respective advantages. In the past few years, numerous works have been proposed in this direction.

One type of integration method is to use each network independently and then combine their results together [119], [121]–[123]. In [123], a parallel framework was proposed to extract spectral–spatial joint features from HSIs. In this framework, SAE was employed to extract spectral features of each pixel, and CNN was used to extract spatial features from the corresponding image patch. These two results were fused by a fully connected layer. Similar to this article, Xu et al. [121] also adopted the parallel framework but used LSTM to extract the spectral features. In contrast, Hang et al. [122] proposed a serial framework to fuse CNNs and RNNs. Specifically, they used a CNN to extract the spatial features from each band of HSIs and then used an RNN to fuse the extracted spatial features. In [119], Shi and Pun also employed a serial framework to integrate the CNN and RNN for spectral–spatial FE.

Another kind of integration method is embedding the core component (i.e., convolutional operators) of CNNs into AEs or RNNs [93], [116]. In [93], an unsupervised spectral–spatial FE network was proposed. The whole framework was similar to AEs, also adopting the so-called encoder–decoder paradigm. However, the fully connected operators in AEs were replaced by convolutional operators, so that the network can directly extract spectral–spatial joint features from cubes. In [116], Liu et al. proposed a spectral–spatial FE method based on a convolutional LSTM network. Instead of fully connected operators, they also used convolutional operators in LSTM units. For a given cube, each band was fed into the convolutional LSTM unit sequentially. The

convolutional operators could extract the spatial features, while the recurrent operators could extract the spectral features. The whole network was optimized in an end-to-end manner, thus achieving satisfactory performance.

## EXPERIMENTAL RESULTS

To evaluate the performance of different FE techniques, we selected four techniques from the UFE category (i.e., PCA [19], multiscale structural total variation (MSTV) [39], OTVCA [42], and LPP [48]), four techniques from the SFE category (i.e., LDA [51], CGDA [74], LSDR [82], and JPlay [87]), and five techniques from the deep FE category [i.e., SAE [124], RNN [122], CNN [125], convolutional AE (CAE) [93], and convolutional RNN (CRNN) [116]]. Here, we set the tuning parameters for those algorithms before representing the experimental results.

### *ALGORITHM SETUP*

The parameter setting usually plays a crucial role in assessing the performance of FE algorithms. Subspace dimension (or number of features, $d$) is a common parameter for all of the compared algorithms. Selection of the number of features is a hard task for HSI analysis. The endmember selection/extraction, subspace identification, and/or rank selection are all referred to this subject [126]–[128]. For a fair and simplified comparison, the parameter $d$ is assigned to be equal to the number of classes ($k$). We should note that $d$ in LDA is automatically determined as $k-1$, due to the class separability (Fisher's criterion).

#### UFE

- *PCA*: This method is a parameter-free technique.
- *MSTV*: In [39], all parameters are adjusted using a trial-and-error approach. The multiscale parameters adjusting the degree of smoothness (as suggested in [39]) are set to 0.003, 0.02, and 0.01. The spatial scale for the structure extraction in three levels (as suggested in [39]) is set to 2, 1, and 3.
- *OTVCA*: This method is initialized as recommended in [42]. The tuning parameter $\lambda$, which controls the level of smoothness applied on the features, is set to 1% of the maximum intensity range of the data sets.
- *LPP*: The number of neighbors is set to 12. The bandwidth of the Gaussian kernel is set to 1.

#### SFE

A common strategy for model selection is to run cross validation (CV) on the training set, since the labeled samples are available in SFE. Therefore, we used the CV strategy on the following studied algorithms for parameter selection.

- *LDA*: This method can be viewed as a baseline for SFE. There is no additional parameter in LDA.
- *CGDA*: Equation (23) can be tuned to a regularized optimization problem, where one extra parameter—regularized $l_2$-norm—must be set in advance in the process of graph construction, which can be searched in the range of $\{10^{-4}, 10^{-3}, 10^{-2}, \ldots, 10^{1}, 10^{2}\}$ by CV. In the experiments, 0.1 is used for all three data sets.
- *LSDR*: Two parameters are involved in LSDR, the standard deviation for the Gaussian function and the regularization parameter, which are selected in the range of $\{0.05, 0.1, \ldots, 0.95, 1\}$ and $\{10^{-2}, 10^{-1}, 10^{1}, 10^{2}\}$, respectively, using CV. Finally, $\sigma$ and $\lambda$ are both set to one in our experiments.
- *JPlay*: There are three regularization parameters ($\alpha$, $\beta$, and $\gamma$) that must be set in the JPlay model (32). With the CV conducted on the training set of three different data sets, the regularization parameters are selected in the ranges of $\{10^{-2}, 10^{-1}, 10^{1}, 10^{2}\}$, yielding the final setting of $(\alpha, \beta, \gamma) = (0.1, 1, 1)$ for the first data set, $(\alpha, \beta, \gamma) = (0.1, 0.1, 1)$ for the second data set, and $(\alpha, \beta, \gamma) = (0.1, 1, 1)$ for the last data set.

#### DEEP FE

- *SAE*: The input of SAE is the original spectral information of each pixel. Three hidden layers are used. The numbers of neurons from the first to the third hidden layer are set to 32, 64, and 128, respectively. Rectified linear unit (ReLU) is adopted as the activation function for each hidden layer.
- *CNN*: The input of CNN is a small cube with a size of $16 \times 16 \times p$, where $p$ represents the number of spectral bands for each data. Three convolutional layers are used. Each convolutional layer is sequentially followed by a batch normalization layer, a ReLU activation function, and a max-pooling layer. Note that the last pooling layer is an adaptive max-pooling layer, making the output size equal to $1 \times 1$ for any input sizes. The kernel size for each convolution is $3 \times 3$, and the numbers of kernels from the first to the third convolutions are set to 32, 64, and 128, respectively. Padding operators are used to preserve the spatial size after each convolutional operator.
- *PCNN*: PCA is applied prior to CNN to reduce the spectral dimension of the HSI. The number of reduced dimensions by PCA is set to the number of classes ($k$). The input cube for the CNN is of size $16 \times 16 \times k$.
- *RNN*: The input of RNN is the same as the input of SAE. Two recurrent layers with GRU are employed. The number of neurons in each recurrent layer is set to 128.
- *Integrated networks*: CAE and CRNN are selected as two representative integrated networks. The input for them is the same as that for the CNN. For CAE, three convolutional layers and three deconvolutional layers are adopted. All of them use $3 \times 3$ kernels. The numbers of kernels from the first to the third convolutional layers are set to 32, 64, and 128, respectively. In contrast, the numbers of kernels from the first and third deconvolutional layers are set to 64, 32, and $p$, respectively. Similar to [116], CRNN adopts two recurrent layers with convolutional LSTM units. For both recurrent layers, $3 \times 3$ convolutional kernels are applied. The numbers of ker-

nels for the first and the second recurrent layers are set to 32 and 64, respectively.

All of these DL-related models are implemented in the PyTorch framework. To optimize them, we use the Adam algorithm with default parameters. The batch size, learning rate, and number of training epochs are set to 128, 0.001, and 200, respectively. To reduce the effects of random initialization, all of the DL models are repeated five times, and the mean values are reported.

### RANDOM FOREST CLASSIFIER

Apart from the deep FE techniques, all of the other FE techniques use random forest (RF) to perform the classification task. The number of trees selected for RF is set to 200. We set the number of the prediction variable approximately to the square root of the number of input bands.

## *PERFORMANCE OF FE TECHNIQUES ON THREE HSIS*

We applied FE techniques on the three hyperspectral data sets—i.e., Indian Pines 2010, Houston 2013, and Houston 2018—and the classification accuracies, including class accuracies, average accuracy (AA), overall accuracy (OA), and kappa coefficient ($\kappa$) are shown in Tables 5, 6, and 7, respectively. The results are first discussed within the categories and then between different categories. We should note that the results and discussions are in terms of classification accuracies obtained from the classification of the HSIs.

### UFE

- *PCA*: PCA demonstrates the poorest performance compared with the other techniques; however, it considerably improves the classification accuracies compared with the results obtained by applying the RF on the spectral bands. One of the main disadvantages of PCA is that it does not take into account the noise; therefore, the extracted features with lower variance are often degraded by different types of noise existing in the HSI [129]. Additionally, PCA takes into account only the spectral correlation, and it entirely neglects the spatial (neighboring) information.
- *LPP*: LPP considerably outperforms the other UFE techniques for the Indian Pines data set. However, in the case of the Houston data sets, it provides very poor results. LPP incorporates the spatial information using the manifold-learning process and by constructing the neighboring graph [48].
- *OTVCA*: OTVCA outperforms the other UFE technqiues for the Houston data sets. In the case of Houston 2013, the improvements are considerable. OTVCA is robust to noise due to the signal model, which takes into account the noise and model errors. Additionally, OTVCA exploits the spatial correlation by incorporating the TV penalty; therefore, the extracted features are piecewise smooth and have a high SNR [42]. Overall, it can be observed that OTVCA, which is a candidate from the

**TABLE 5. THE CLASSIFICATION ACCURACIES OBTAINED ON FEATURES EXTRACTED FROM THE INDIAN PINES 2010 DATA SET USING DIFFERENT SHALLOW AND DEEP FE TECHNIQUES.**

| | INDIAN PINES 2010 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SHALLOW FE | | | | | | | | DEEP FE | | | | | |
| | | UFE | | | | SFE | | | | SFE | | | | | |
| | SPECTRAL | PCA | MSTV | OTVCA | LPP | LDA | CGDA | LSDR | JPLAY | SAE | RNN | CNN | CAE | CRNN | PCNN |
| 1 | 0.926 | 0.9064 | **0.9992** | 0.926 | 0.885 | 0.9628 | 0.8144 | 0.8459 | 0.923 | 0.9327 | 0.8829 | 0.9275 | 0.9432 | 0.959 | 0.9397 |
| 2 | 0.8769 | 0.9976 | **1** | **1** | 0.9976 | **1** | 0.9961 | 0.8933 | 0.9984 | 0.9573 | 0.9178 | 0.9544 | 0.9961 | 0.8596 | 0.9998 |
| 3 | 0.8862 | 0.9724 | 0.9724 | 0.9897 | 0.9724 | 0.9724 | 0.9759 | 0.9655 | 0.9862 | 0.9683 | 0.9405 | 0.989 | **0.9986** | 0.8241 | 0.9938 |
| 4 | 0.6888 | 0.7762 | **1** | 0.8953 | 0.8742 | 0.927 | 0.7474 | 0.8194 | 0.83 | 0.7508 | 0.7621 | 0.884 | 0.8822 | 0.8569 | 0.893 |
| 5 | 0.8058 | **0.8855** | 0.8039 | 0.8151 | 0.8682 | 0.8802 | 0.8096 | 0.8394 | 0.8665 | 0.8488 | 0.8474 | 0.8692 | 0.857 | 0.8638 | 0.8706 |
| 6 | 0.8172 | 0.8797 | **0.9946** | 0.7094 | 0.9739 | 0.7883 | 0.8284 | 0.9397 | 0.9418 | 0.9013 | 0.9204 | 0.9268 | 0.9167 | 0.6975 | 0.9127 |
| 7 | 0.417 | 0.5954 | 0.6792 | **0.7166** | 0.6985 | **0.7166** | 0.6845 | 0.703 | 0.6958 | 0.6265 | 0.5795 | 0.6507 | 0.6818 | 0.6348 | 0.7103 |
| 8 | 0.253 | 0.2583 | 0.2599 | 0.2768 | 0.2961 | 0.2955 | 0.2431 | 0.2952 | 0.2758 | 0.5349 | 0.284 | 0.8776 | 0.6776 | **0.9934** | 0.4725 |
| 9 | 0.6545 | 0.7498 | 0.7048 | 0.7943 | **0.8913** | 0.8452 | 0.7971 | 0.8419 | 0.8142 | 0.8732 | 0.8533 | 0.8302 | 0.8336 | 0.8194 | 0.8621 |
| 10 | 0.8229 | 0.9406 | 0.9594 | 0.9368 | 0.9019 | **0.9804** | 0.8514 | 0.7761 | 0.9663 | 0.9015 | 0.8096 | 0.8368 | 0.8752 | 0.7946 | 0.9289 |
| 11 | 0.6658 | 0.8402 | 0.9224 | **0.9945** | 0.8943 | 0.9288 | 0.7195 | 0.7651 | 0.8052 | 0.8121 | 0.7414 | 0.744 | 0.7633 | 0.6492 | 0.9165 |
| 12 | 0.9985 | **1** | **1** | **1** | 0.9995 | **1** | **1** | **1** | **1** | 0.9998 | 0.9765 | 0.9945 | 0.9838 | 0.9748 | 0.9991 |
| 13 | 0.9468 | **0.9962** | 0.9879 | 0.9888 | 0.9925 | 0.983 | 0.958 | 0.9738 | 0.9819 | 0.9621 | 0.9427 | 0.9925 | 0.993 | 0.9892 | 0.9959 |
| 14 | 0.8783 | 0.9 | 0.9615 | 0.9145 | 0.9344 | 0.9174 | 0.8756 | 0.8628 | 0.8953 | 0.9094 | 0.903 | 0.9984 | 0.9985 | 0.8981 | **0.9993** |
| 15 | 0.9333 | 0.9667 | 0.9511 | 0.9933 | 0.9489 | 0.9756 | 0.9333 | 0.9311 | 0.96 | 0.9307 | 0.8119 | 0.9947 | 0.9942 | 0.7556 | **0.9978** |
| 16 | 0.3735 | 0.2036 | 0.2885 | 0.1601 | 0.5217 | 0.1719 | 0.3439 | 0.4901 | 0.4466 | 0.2053 | 0.206 | 0.098 | 0.17 | **0.6028** | 0.1051 |
| AA | 0.7465 | 0.8043 | 0.8428 | 0.8194 | **0.8532** | 0.8341 | 0.7861 | 0.8089 | 0.8367 | 0.8197 | 0.7737 | 0.848 | 0.8478 | 0.8233 | 0.8498 |
| OA | 0.7866 | 0.8598 | 0.8561 | 0.8378 | **0.9112** | 0.8748 | 0.837 | 0.8748 | 0.8829 | 0.8836 | 0.8655 | 0.8945 | 0.8911 | 0.8525 | 0.9018 |
| $\kappa$ | 0.739 | 0.8297 | 0.8247 | 0.8054 | **0.8909** | 0.8481 | 0.801 | 0.8466 | 0.8571 | 0.858 | 0.8355 | 0.8716 | 0.8673 | 0.8213 | 0.8802 |

The highest accuracy in each row is shown in bold.

**TABLE 6. THE CLASSIFICATION ACCURACIES OBTAINED ON FEATURES EXTRACTED FROM THE HOUSTON UNIVERSITY 2013 DATA SET USING DIFFERENT SHALLOW AND DEEP FE TECHNIQUES.**

| | HOUSTON 2013 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SHALLOW FE | | | | | | | | DEEP FE | | | | | |
| | | UFE | | | | SFE | | | | SFE | | | | | |
| | SPECTRAL | PCA | MSTV | OTVCA | LPP | LDA | CGDA | LSDR | JPLAY | SAE | RNN | CNN | CAE | CRNN | PCNN |
| 1 | 0.8262 | **0.8272** | 0.8025 | 0.8205 | 0.811 | 0.8177 | 0.8139 | 0.812 | 0.7768 | 0.8217 | 0.8182 | 0.8104 | 0.8154 | 0.8245 | 0.8089 |
| 2 | 0.8318 | 0.8393 | 0.8412 | 0.8515 | 0.8214 | 0.8355 | 0.8327 | 0.8553 | **0.9662** | 0.8274 | 0.8153 | 0.8425 | 0.8167 | 0.8412 | 0.8293 |
| 3 | 0.9782 | **1** | 0.9822 | **1** | **1** | **1** | **1** | **1** | 0.998 | 0.9895 | 0.9939 | 0.8594 | 0.7731 | 0.9156 | 0.8432 |
| 4 | 0.9138 | 0.9081 | 0.7633 | 0.8873 | 0.9479 | 0.892 | 0.9053 | 0.8864 | 0.9564 | **0.9773** | 0.904 | 0.917 | 0.9153 | 0.9129 | 0.9159 |
| 5 | 0.9659 | 0.9886 | 0.9915 | **0.9991** | 0.9867 | 0.9384 | 0.9915 | 0.9688 | 0.9782 | 0.9438 | 0.9389 | 0.9699 | 0.9585 | 0.9881 | 0.9824 |
| 6 | 0.9930 | 0.993 | 0.958 | 0.958 | 0.979 | **1** | 0.8741 | 0.986 | 0.993 | 0.9874 | 0.9678 | 0.8769 | 0.9776 | 0.9483 | 0.9497 |
| 7 | 0.7463 | 0.8927 | 0.6362 | 0.709 | **0.9123** | 0.7901 | 0.8535 | 0.8526 | 0.7817 | 0.7293 | 0.7392 | 0.8802 | 0.8694 | 0.8642 | 0.8627 |
| 8 | 0.3305 | 0.4606 | 0.5992 | 0.6724 | 0.4311 | 0.7379 | 0.4302 | 0.471 | 0.7806 | 0.3792 | 0.4153 | 0.6344 | 0.6762 | 0.5305 | **0.8351** |
| 9 | 0.6771 | 0.7885 | 0.8706 | **0.9008** | 0.7413 | 0.6449 | 0.7186 | 0.6752 | 0.7592 | 0.7145 | 0.7367 | 0.8595 | 0.854 | 0.8404 | 0.8691 |
| 10 | 0.4295 | 0.4749 | 0.6612 | **0.8398** | 0.4595 | 0.4662 | 0.4826 | 0.5792 | 0.6014 | 0.5556 | 0.5373 | 0.5674 | 0.5782 | 0.4514 | 0.6168 |
| 11 | 0.7011 | 0.7268 | 0.982 | **0.9924** | 0.7306 | 0.7239 | 0.7287 | 0.5806 | 0.6983 | 0.6231 | 0.725 | 0.7417 | 0.7292 | 0.6186 | 0.7913 |
| 12 | 0.5485 | 0.9145 | 0.7349 | **0.9625** | 0.756 | 0.6513 | 0.7656 | 0.5687 | 0.7858 | 0.6305 | 0.7606 | 0.9379 | 0.9402 | 0.844 | 0.9593 |
| 13 | 0.614 | 0.7754 | 0.6982 | 0.7789 | 0.8105 | 0.6105 | 0.7719 | 0.6702 | 0.7509 | 0.4516 | 0.6656 | 0.8835 | **0.8968** | 0.8414 | 0.8765 |
| 14 | 0.9838 | 0.9919 | **1** | **1** | 0.996 | 0.9919 | 0.9879 | 0.9595 | 0.9879 | 0.9692 | 0.985 | 0.9943 | 0.9773 | 0.9603 | 0.9968 |
| 15 | 0.9789 | 0.9746 | **1** | 0.9789 | 0.9746 | 0.9831 | 0.9852 | 0.9514 | 0.9831 | 0.9732 | 0.9607 | 0.8072 | 0.7471 | 0.9345 | 0.8592 |
| AA | 0.7679 | 0.8371 | 0.8347 | **0.8901** | 0.8239 | 0.8056 | 0.8094 | 0.7878 | 0.8532 | 0.7716 | 0.7976 | 0.8388 | 0.835 | 0.821 | 0.8664 |
| OA | 0.7278 | 0.8058 | 0.8088 | **0.8753** | 0.7874 | 0.7745 | 0.7789 | 0.7524 | 0.828 | 0.7436 | 0.7646 | 0.8239 | 0.8184 | 0.7921 | 0.8526 |
| $\kappa$ | 0.7076 | 0.7895 | 0.7923 | **0.8648** | 0.77 | 0.7552 | 0.7604 | 0.7315 | 0.8134 | 0.7235 | 0.7469 | 0.8096 | 0.8036 | 0.7761 | 0.8404 |

The highest accuracy in each row is shown in bold.

low-rank reconstruction techniques, generally provides better classification accuracies than the other UFE techniques.

### SFE

- *LDA versus spectral classifier (RF)*: With the embedding of supervised information, LDA obviously performs better than the situation where RF is directly applied to the spectral signatures, in terms of the overall performance and individual accuracies for most materials. This indicates the effectiveness of SFE to a great extent.
- *LDA versus CGDA*: Although the classification performance of CGDA is inferior to that of LDA from an overall perspective, the advantage of CGDA mainly lies in its automation in computing the similarity (or connectivity) between samples. This could lead to a relatively stable FE, particularly in large-scale and more complex hyperspectral scenes. Due to the data-driven graph embedding, CGDA yields a lower running speed than LDA in the process of model training.
- *LDA versus LSDR*: Intuitively, LSDR provides competitive classification performance with LDA. However, LSDR is time consuming due to the distribution matching between input samples and labels. The requirement to estimate the statistical distribution also limits LSDR's stability, especially when the training set is available on a small scale (e.g., for the Indian Pines 2010 and Houston 2013 data sets).
- *LDA versus JPlay*: Unlike conventional regression techniques, JPlay is capable of extracting semantically meaningful and robust features, due to the multilayered structure and self-reconstruction constraint (32). Quantitatively speaking, JPlay outperforms the other SFE methods. The CV provides a feasible solution to automatically determine the parameter combination in JPlay. Despite the ADMM solver designed for speeding up the optimization process, such a multilayered parameter update inevitably suffers from high computational cost.

### DEEP FE

- *Spectral versus spectral–spatial models*: Most of the spectral–spatial models (i.e., CNN, PCNN, CAE, and CRNN) achieve superior performance compared to spectral models (i.e., SAE and RNN) in terms of AA, OA, and kappa due to the joint use of spectral and spatial information. This indicates that, besides the rich spectral information, spatial information is also important for HSI classification.
- *PCNN and CNN versus CAE and CRNN*: Similar to SAE, CAE focuses on image reconstruction rather than classification. In contrast, PCNN and CNN are exclusively designed for the classification task, so they are able to learn more discriminative features than CAE, leading to better classification performance, especially on the Houston 2018 data set. Although CRNN also focuses on the classification task, it has more parameters to train. Using the same number of training samples and epochs,

**TABLE 7. THE CLASSIFICATION ACCURACIES OBTAINED ON FEATURES EXTRACTED FROM THE HOUSTON UNIVERSITY 2018 DATA SET USING DIFFERENT SHALLOW AND DEEP FE TECHNIQUES.**

| | HOUSTON 2018 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SHALLOW FE | | | | | | | | DEEP FE | | | | | |
| | | UFE | | | | SFE | | | | SFE | | | | | |
| | SPECTRAL | PCA | MSTV | OTVCA | LPP | LDA | CGDA | LSDR | JPLAY | SAE | RNN | CNN | CAE | CRNN | PCNN |
| 1 | 0.3088 | **0.8781** | 0.0536 | 0.6842 | 0.6618 | 0.6256 | 0.7575 | 0.7969 | 0.5991 | 0.794 | 0.5702 | 0.7516 | 0.4428 | 0.6338 | 0.6638 |
| 2 | 0.7603 | 0.8396 | 0.7046 | 0.6376 | 0.8122 | 0.8474 | 0.8076 | 0.7747 | 0.8347 | 0.7893 | 0.6975 | 0.8173 | **0.8849** | 0.8707 | 0.8376 |
| 3 | **1** | **1** | **1** | 0.9972 | **1** | **1** | **1** | **1** | **1** | **1** | 0.9972 | 0.7739 | 0.8482 | 0.9924 | 0.8045 |
| 4 | 0.9134 | 0.9494 | 0.6238 | 0.6775 | 0.9453 | 0.9059 | 0.9265 | 0.9276 | 0.9298 | 0.9221 | 0.8613 | 0.9444 | 0.9362 | 0.9439 | **0.9595** |
| 5 | 0.4119 | 0.4668 | 0.2676 | 0.1679 | 0.4728 | 0.5258 | 0.4661 | 0.4289 | 0.3971 | 0.4982 | 0.404 | 0.433 | 0.5396 | **0.5404** | 0.48 |
| 6 | 0.257 | 0.299 | **0.3835** | 0.3164 | 0.3008 | 0.291 | 0.2776 | 0.2726 | 0.278 | 0.2585 | 0.2537 | 0.305 | 0.308 | 0.2902 | 0.3377 |
| 7 | 0.3025 | 0.3025 | 0.3025 | 0.3025 | 0.3025 | 0.3025 | 0.3109 | 0.3025 | 0.2857 | 0.3025 | 0.3025 | 0.2908 | 0.2723 | 0.2997 | **0.3176** |
| 8 | 0.7657 | 0.7675 | 0.7599 | 0.7417 | 0.7785 | 0.7849 | 0.7544 | 0.7518 | 0.7771 | 0.7216 | 0.7356 | 0.8538 | 0.8583 | 0.8092 | **0.8677** |
| 9 | 0.3849 | 0.3877 | 0.5767 | 0.599 | 0.4887 | 0.3917 | 0.3672 | 0.5255 | 0.5877 | 0.6302 | 0.4186 | 0.7970 | 0.7371 | 0.3717 | **0.8659** |
| 10 | 0.3603 | 0.436 | 0.3747 | 0.4491 | 0.423 | 0.4086 | 0.379 | 0.3497 | 0.401 | 0.3819 | 0.3465 | **0.5902** | 0.4957 | 0.5484 | 0.5778 |
| 11 | 0.4162 | 0.4792 | **0.7862** | 0.7596 | 0.5085 | 0.4667 | 0.4266 | 0.4422 | 0.5359 | 0.4143 | 0.4699 | 0.5456 | 0.5781 | 0.6048 | 0.5948 |
| 12 | 0.0132 | 0.0046 | 0.0093 | 0.0077 | 0.007 | 0.0023 | 0.017 | 0 | 0.0302 | 0.0152 | 0.0697 | 0.0511 | 0.0477 | **0.0927** | 0.0579 |
| 13 | 0.4525 | 0.5556 | 0.4238 | 0.409 | 0.5442 | 0.5164 | 0.5707 | 0.5603 | 0.5324 | 0.4523 | 0.4789 | 0.5148 | 0.5619 | 0.4246 | **0.5811** |
| 14 | 0.3019 | 0.2629 | 0.546 | 0.4665 | 0.3651 | 0.4152 | 0.2294 | 0.2073 | 0.3212 | 0.3789 | 0.3309 | 0.5289 | **0.6763** | 0.3375 | 0.5705 |
| 15 | 0.6303 | 0.4721 | 0.4457 | 0.4887 | 0.4602 | 0.5549 | 0.418 | 0.5234 | 0.5944 | 0.5197 | 0.5289 | 0.6277 | 0.6476 | 0.6447 | **0.6591** |
| 16 | 0.6412 | 0.7611 | 0.622 | 0.7648 | 0.7559 | 0.5888 | 0.7374 | 0.6403 | 0.6688 | 0.7457 | 0.7086 | 0.8498 | 0.7594 | 0.7173 | **0.8572** |
| 17 | **1** | **1** | **1** | **1** | **1** | **1** | 0.9545 | **1** | **1** | **1** | **1** | 0.9545 | 0.8909 | **1** | **1** |
| 18 | 0.4983 | 0.6885 | 0.6576 | 0.5197 | 0.714 | 0.6581 | 0.6625 | 0.5686 | 0.7366 | 0.5346 | 0.608 | 0.6365 | 0.5981 | **0.7692** | 0.702 |
| 19 | 0.5265 | 0.6363 | 0.906 | 0.8777 | 0.7323 | 0.6989 | 0.61 | 0.6266 | 0.6771 | 0.6569 | 0.6545 | 0.9102 | 0.896 | 0.8006 | **0.9476** |
| 20 | 0.4444 | 0.8904 | **0.9706** | 0.5253 | 0.8479 | 0.9189 | 0.6797 | 0.5955 | 0.5393 | 0.5388 | 0.4801 | 0.6246 | 0.5566 | 0.4879 | 0.6519 |
| AA | 0.5195 | 0.6039 | 0.5707 | 0.5696 | 0.606 | 0.5952 | 0.5676 | 0.5647 | 0.5863 | 0.5777 | 0.5458 | 0.64 | 0.6268 | 0.609 | **0.6667** |
| OA | 0.4634 | 0.5101 | 0.575 | 0.5899 | 0.5552 | 0.5027 | 0.4825 | 0.5492 | 0.5944 | 0.5938 | 0.4851 | 0.7278 | 0.6969 | 0.5116 | **0.7728** |
| $\kappa$ | 0.3732 | 0.4317 | 0.4833 | 0.4974 | 0.4714 | 0.4231 | 0.4018 | 0.456 | 0.5037 | 0.4948 | 0.3936 | 0.6474 | 0.6124 | 0.4372 | **0.7011** |

The highest accuracy in each row is shown in bold.

PCNN and CNN can achieve better results than CRNN in terms of AA, OA, and kappa.

- *PCNN versus CNN*: PCNN outperforms CNN in terms of classification accuracies for all three data sets. We should note that the improvements are substantial in the case of the Houston 2013 and 2018 data sets. Due to the use of PCA, most of the redundant spectral information is reduced. Therefore, the number of trainable parameters in PCNN is smaller than that of CNN, making it easier to learn under the same condition.

### SHALLOW UFE VERSUS SHALLOW SFE

For all three data sets used in the experiments, the UFE techniques provide better classification accuracies than the SFE techniques. Unlike SFE, UFE tends to pay more attention to spatial–spectral information extraction because it fully considers all samples of HSI as the model input. Conversely, the performance of SFE is, to a great extent, limited by the ability to largely gather HSI ground sampling. Direct evidence is given in Tables 5–7. For the Indian Pines 2010 and Houston 2018 data sets, for which more training samples are available, SFE-based methods produce results competitive with those of UFE-based techniques, whereas for the Houston 2013 data set, the classification performance of SFE is relatively inferior to that of UFE, due to the small-scale training set. 2013 Considering the low number of ground samples often available in HSI applications, the experimental results confirm the advantage of UFE over SFE for HSI FE.

### SHALLOW FE VERSUS DEEP FE

At first glance, the shallow FE approaches slightly outperform the deep FE techniques for the two data sets, i.e., Indian Pines 2010 and Houston 2013. However, a deep comparison reveals that some deep FE techniques, such as CNN-based FE, provide consistency and good performance over all three data sets. Additionally, when the dimension-reduced step (e.g., using PCA) is applied prior to the CNN technique, the resulting PCNN yields, by far, the second highest accuracies in the case of the Indian Pines 2010 and Houston 2013 data sets (only moderately lower than LPP or OTVCA, respectively) and simultaneously obtains the best performance on the Houston 2018 data set.

It is worth mentioning that CNN-based FE methods obtained at least a 10% increase over the shallow techniques in the case of the Houston 2018 data set. This could be due

to the high nonlinear behavior of this data set, which contains 20 land cover classes. The main factors for CNN-based FE methods to obtain approximately 20% improvement over the shallow FE methods on the Houston 2018 data set are the availability of sufficient training samples and modeling the spatial information of the HSI well.

## COMPARISONS OF THE LAND COVER MAPS

Figures 10–12 compare the classification maps for the Indian Pines, Houston University 2013, and Houston University 2018 data sets, respectively. The figures compare the maps obtained from methods that provide the highest OA from each category (i.e., shallow UFE, shallow SFE, and deep FE) along with the map obtained from the spectral classifier (HSI). Additionally, we depict the maps obtained by CNN for all three data sets since this provides the highest OA among the deep FE techniques, which do not exploit a reduction step.

Overall, the classification maps of either the UFE- or SFE-based approaches (e.g., LPP, JPlay, CNN, PCNN) are smoother compared to HSI, which tends to generate sparse mislabeled pixels. More specifically, the classification maps generated by spectral–spatial FE-based methods, e.g., OTVCA, CNN, and PCNN, are usually a bit oversmoothed, leading to the creation of fake structures, especially for the Indian Pines 2010 and Houston 2018 data sets. In the case of OTVCA, the oversmoothing can be avoided by decreasing the tuning parameter. In contrast, JPlay obtains relatively desirable classification maps, despite the lack of spatial information modeling. It is worth mentioning that the JPlay algorithm can maintain the structural information for the Houston 2013 data set in the shadow-covered region, where pixels at some bands are considerably attenuated. This is due to the elimination of the spectral variability using self-reconstruction regularization [the third term in (32)] and the multilayered linearized regression technique.

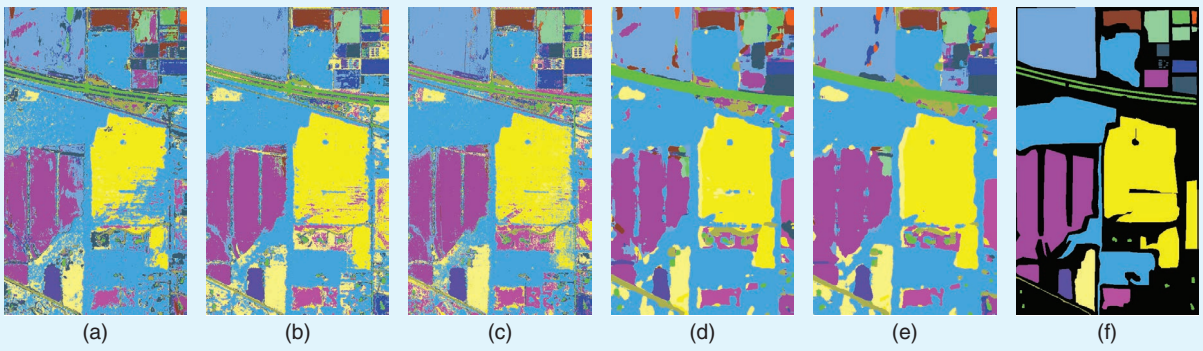


**FIGURE 10.** The classification maps obtained on the extracted features from the Indian Pines 2010 data set: (a) HSI, (b) LPP, (c) JPlay, (d) CNN, (e) PCNN, and (f) ground reference. From each category, the method with the highest OA is shown for the demonstration, and (a) is the one obtained from the spectral bands.

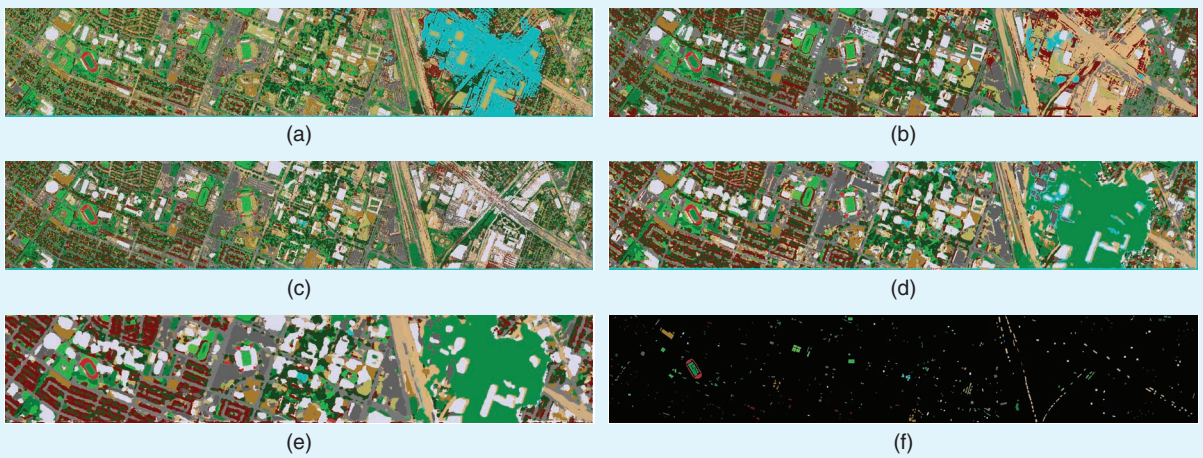


**FIGURE 11.** The classification maps obtained on the extracted features from Houston University 2013 data set: (a) HSI, (b) OTVCA, (c) JPlay, (d) CNN, (e) PCNN, and (f) ground reference. From each category, the method with the highest OA is shown for the demonstration, and (a) is the one obtained from the spectral bands.


**FIGURE 12.** The classification maps obtained on the extracted features from Houston University 2018 data set: (a) HSI, (b) OTVCA, (c) JPlay, (d) CNN, (e) PCNN, and (f) ground reference. From each category, the method with the highest OA is shown for the demonstration, and (a) is the one obtained from the spectral bands.

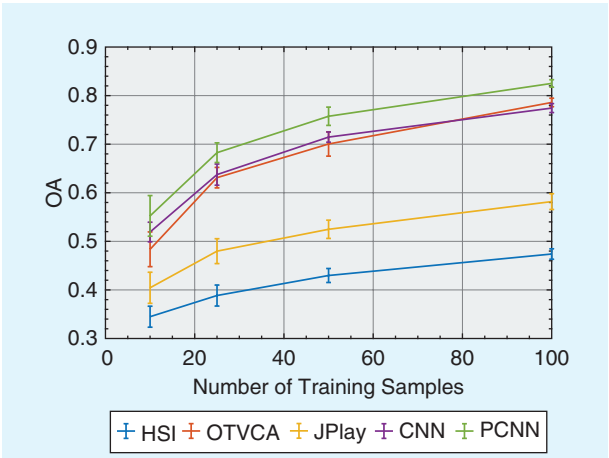


**FIGURE 13.** The classification accuracies with respect to the number of the training samples on the Houston University 2018 data set. The results shown are means over 10 experiments, and standard deviations are shown by the error bars.

### *PERFORMANCE WITH RESPECT TO THE NUMBER OF TRAINING SAMPLES*

In this section, we investigate the performance of the FE techniques in terms of classification accuracies with respect to the number of training samples. As we have already stated, this analysis is of great interest for two main reasons. First, ground sample acquisition and measurements are often cumbersome and could be impossible in cases for which the target area is not reachable. Additionally, the limited number of samples affects the performance of not only the supervised classifiers but also the SFE techniques, since they are highly reliant on the number of training samples. Therefore, in this experiment, we perform an analysis on the 2017 Houston University data set by comparing the performances of the FE techniques when selecting 10, 25, 50, and 100 training samples randomly. Figure 13 compares the OAs obtained by applying RF on the spectral bands (labeled by HSI) and the features extracted by OTVCA and JPlay along with the OAs obtained by CNN and PCNN. The results are mean values over 10 experiments based on selecting the samples randomly. (The standard deviations are shown by the error bars.) The outcomes of the experiment can be summarized as follows:

- The SFE technique (i.e., JPlay) improves the OAs compared to the spectral classifier. However, it provides a much lower OA compared with UFE and deep FE for all cases. Two aspects might explain this point. One is that JPlay fails to model spatial and contextual information; another is that, although JPlay attempts to enhance the reorientation ability of the features via multilayered linear mapping, it is still incomparable to the nonlinear deep-FE-based techniques, particularly when the number of samples is increased.
- In this experiment, the UFE technique (i.e. OTVCA) and the deep FE method, CNN, performed similarly in terms of classification accuracies. Compared with the results given in Table 7, it can be observed that the random selection of the training samples over the entire class of regions from the ground reference considerably improves the performance of RF applied on the features extracted by OTVCA. This is often due to the lack of a parameter selection technique to choose the optimum parameter

for the OTVCA algorithm, which could lead to oversmoothing on the features.

- The DL technique (i.e., PCNN), after using the reduction (i.e., PCA), provides very high OA for all the cases. Comparing the results with CNN (i.e., without using the PCA reduction) confirms the advantage of using the reduction stage prior to DL techniques.

## CONCLUSIONS AND SUMMARY

In the past decade, HSI FE has considerably evolved, leading to three main research lines (i.e., shallow UFE, shallow SFE, and deep FE approaches) that include the majority of FE techniques presented in this article. We systematically provided a technical overview of the state-of-the-art techniques proposed in the literature by categorizing the aforementioned three focuses into subcategories. To make this research article easy to follow for researchers at different levels (i.e., students, researchers, and senior researchers), we aimed to show the evolution of each category over the decades rather than including many techniques with an exhaustive reference list.

The experimental section was designed to compare the performances of the techniques in two ways: 1) between all of the categories (i.e., shallow UFE, shallow SFE, and deep FE approaches) and 2) within each category by analyzing the corresponding subcategories. In this manner, a various subcategories were investigated, detailing the evolution of the shallow UFE (i.e., conventional data-projection schemes, band clustering/splitting techniques, low-rank reconstruction techniques, and manifold-learning techniques), shallow SFE (i.e., class-separation discriminant analysis, graph-embedding discriminant analysis, regression-based representation learning, and JPlay), and deep FE approaches (i.e., AE, CNN, RNN, and integrative approaches). Three recent hyperspectral data sets were studied, and the results were evaluated in terms of classification accuracies and the quality of the classification maps.

The experiments carried out in this study showed the following, in terms of classification accuracies: 1) DL FE techniques (i.e., CNN and PCNN) can outperform the shallow methods, particularly when a sufficient amount of training data are available; 2) applying a dimensionality reduction step (such as PCA) prior to the DL techniques considerably improves their performances; and 3) shallow UFE techniques not only outperform the SFE methods but also are very competitive compared with deep FE methods. However, we should mention that the conclusions are limited by the experiments carried out on the three HSI data sets. In addition, this article provides an impressive amount of code and libraries, mostly written in Python and MATLAB, to ease the task of researchers in this vibrant field of research.

## ACKNOWLEDGMENTS

We would like to thank Prof. Melba Crawford for providing the Indian Pines 2010 Data and the National Center for Airborne Laser Mapping, the Hyperspectral Image Analysis Laboratory at the University of Houston, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee. This work is partially supported by an Alexander von Humboldt research grant. We also would like to thank the AXA Research Fund for supporting the work of Prof. Jocelyn Chanussot and the corresponding author of this paper, Dr. Danfeng Hong.

## AUTHOR INFORMATION

**Behnood Rasti** (behnood.rasti@gmail.com) is with the Machine Learning Group, Exploration Division, Helmholtz Institute Freiberg for Resource Technology, Helmholtz-Zentrum Dresden-Rossendorf, Germany. He is a Senior Member of the IEEE.

**Danfeng Hong** (hongdanfeng1989@gmail.com) is with the Université Grenoble Alpes, Centre National de la Recherche Scientifique, Grenoble Institute of Technology, Grenoble Images Parole Signal Automatique-Lab, France, and also with the Remote Sensing Technology Institute, German Aerospace Center, Weßling, Germany. He is a Member of the IEEE.

**Renlong Hang** (renlong_hang@163.com) is with the Jiangsu Key Laboratory of Big Data Analysis Technology, School of Automation, Nanjing University of Information Science and Technology, China. He is a Member of the IEEE.

**Pedram Ghamisi** (p.ghamisi@gmail.com) is with the Machine Learning Group, Exploration Division, Helmholtz Institute Freiberg for Resource Technology, Helmholtz-Zentrum Dresden-Rossendorf, Germany. He is a Senior Member of the IEEE.

**Xudong Kang** (xudong_kang@163.com) is with the College of Electrical and Information Engineering, Hunan University, Changsha, China, and Key Laboratory of Visual Perception and Artificial Intelligence of Hunan Province, Changsha, China. He is a Senior Member of the IEEE.

**Jocelyn Chanussot** (jocelyn@hi.is) is with the Université Grenoble Alpes, INRIA, Centre National de la Recherche Scientifique, Grenoble Institute of Technology, Laboratoire Jean Kuntzmann, France, and the faculty of Electrical and Computer Engineering, University of Iceland, Reykjavik. He is a Fellow of the IEEE.

**Jon Atli Benediktsson** (benedikt@hi.is) is with the faculty of Electrical and Computer Engineering, University of Iceland, Reykjavik. He is a Fellow of the IEEE.

## REFERENCES

[1] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 5, no. 1, pp. 8–32, Mar. 2017. doi: 10.1109/MGRS.2016.2616418.

[2] D. W. Scott, *Multivariate Density Estimation*. New York: Wiley, 1992.

[3] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *J. Am. Stat. Assoc.*, vol. 85, no. 411, pp. 664–675, 1990. doi: 10.1080/01621459.1990.10474926.

[4] L. Jimenez and D. Landgrebe, "Supervised classification in high-dimensional space: Geometrical, statistical, and as-

ymptotical properties of multivariate data," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 28, no. 1, pp. 39–54, 1998. doi: 10.1109/5326.661089.

[5] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Boston: Artech House, 2015.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic, 1990.

[7] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley, 2003.

[8] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, 1968. doi: 10.1109/TIT.1968.1054102.

[9] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Comput. Vis.*, vol. 3, no. 4, p. 213, 2009. doi: 10.1049/iet-cvi.2009.0034.

[10] L. Guanter et al., "The EnMAP spaceborne imaging spectroscopy mission for earth observation," *Remote Sens.*, vol. 7, no. 7, pp. 8830–8857, July 2015. doi: 10.3390/rs70708830.

[11] P. Ghamisi et al., "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 6, no. 3, pp. 10–43, Sept. 2018. doi: 10.1109/MGRS.2018.2854840.

[12] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan 2018. doi: 10.1109/TCYB.2016.2605044.

[13] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, 2013. doi: 10.1109/JPROC.2012.2197589.

[14] W. Li, F. Feng, H. Li, and Q. Du, "Discriminant analysis-based dimension reduction for hyperspectral image classification: A survey of the most recent advances and an experimental comparison of different techniques," *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 6, no. 1, pp. 15–34, Mar. 2018. doi: 10.1109/MGRS.2018.2793873.

[15] X. Jia, B.-C. Kuo, and M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013. doi: 10.1109/JPROC.2012.2229082.

[16] W. Sun and Q. Du, "Hyperspectral band selection: A review," *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 7, no. 2, pp. 118–139, June 2019. doi: 10.1109/MGRS.2019.2911100.

[17] J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012. doi: 10.1109/JSTARS.2012.2194696.

[18] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015. doi: 10.1109/TGRS.2014.2358934.

[19] I. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[20] J. Senthilnath, S. N. Omkar, V. Mani, N. Karnwal, and S. P. B, "Crop stage classification of hyperspectral data using unsupervised techniques," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 861–866, Apr. 2013. doi: 10.1109/JSTARS.2012.2217941.

[21] A. Green, M. Berman, P. Switzer, and M. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geos. Remote Sens.*, vol. 26, no. 1, pp. 65–74, 1988. doi: 10.1109/36.3001.

[22] J. Lee, A. Woodyatt, and M. Berman, "Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform," *IEEE Trans. Geos. Remote Sens.*, vol. 28, no. 3, pp. 295–304, 1990. doi: 10.1109/36.54356.

[23] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. New York: Wiley, 2001.

[24] A. Villa, J. Chanussot, C. Jutten, J. Benediktsson, and S. Moussaoui, "On the use of ICA for hyperspectral image analysis," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, July 2009, vol. 4, pp. IV-97–IV-100, doi: 10.1109/IGARSS.2009.5417363.

[25] A. A. Nielsen, "Kernel maximum autocorrelation factor and minimum noise fraction transformations," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 612–624, Mar. 2011. doi: 10.1109/TIP.2010.2076296.

[26] F. Mei, C. Zhao, L. Wang, and H. Huo, "Anomaly detection in hyperspectral imagery based on kernel ICA feature extraction," in *Proc. 2008 2nd Int. Symp. Intelligent Information Technology Application (IITA '08)*, vol. 1. Washington, D.C.: IEEE Computer Society, 2008, pp. 869–873. doi: 10.1109/IITA.2008.98.

[27] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[28] S. Marchesi and L. Bruzzone, "ICA and kernel ICA for change detection in multispectral remote sensing images," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, July 2009, vol. 2, pp. II-980–II-983. doi: 10.1109/IGARSS.2009.5418265.

[29] M. Fauvel, J. Chanussot, and J. Benediktsson, "Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, p. 783,194, 2009. doi: 10.1155/2009/783194.

[30] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2331–2338, Oct. 2002. doi: 10.1109/TGRS.2002.804721.

[31] S. Kumar, J. Ghosh, and M. M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1368–1379, July 2001. doi: 10.1109/36.934070.

[32] A. Martnez-UsMartinez-Uso, F. Pla, J. M. Sotoca, and P. Garca-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007. doi: 10.1109/TGRS.2007.904951.

[33] C. Cariou, K. Chehdi, and S. Le Moan, "Bandclust: An unsupervised band reduction method for hyperspectral remote sensing," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 565–569, May 2011. doi: 10.1109/LGRS.2010.2091673.

[34] S. Rashwan and N. Dobigeon, "A split-and-merge approach for hyperspectral band selection," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1378–1382, Aug 2017. doi: 10.1109/LGRS.2017.2713462.

[35] X. Kang, S. Li, and J. A. Benediktsson, "Feature extraction of hyperspectral images with image fusion and recursive filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3742–3752, June 2014. doi: 10.1109/TGRS.2013.2275613.

[36] X. Kang, S. Li, L. Fang, and J. A. Benediktsson, "Intrinsic image decomposition for feature extraction of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2241–2253, Apr. 2015. doi: 10.1109/TGRS.2014.2358615.

[37] X. Jin, Y. Gu, and T. Liu, "Intrinsic image recovery from remote sensing hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 224–238, Jan. 2019. doi: 10.1109/TGRS.2018.2853178.

[38] X. Jin and Y. Gu, "Superpixel-based intrinsic image decomposition of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4285–4295, Aug. 2017. doi: 10.1109/TGRS.2017.2690445.

[39] P. Duan, X. Kang, S. Li, and P. Ghamisi, "Noise-robust hyperspectral image classification via multi-scale total variation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1948–1962, Jun. 2019. doi: 10.1109/JSTARS.2019.2915272.

[40] B. Rasti, "Sparse hyperspectral image modeling and restoration," Ph.D. dissertation, Dep. Elect. Comput. Eng., Univ. of Iceland, Reykjavík, Dec. 2014.

[41] B. Rasti, J. Sveinsson, and M. Ulfarsson, "Wavelet-based sparse reduced-rank regression for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6688–6698, Oct. 2014. doi: 10.1109/TGRS.2014.2301415.

[42] B. Rasti, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral feature extraction using total variation component analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 6976–6985, Dec. 2016. doi: 10.1109/TGRS.2016.2593463.

[43] B. Rasti, P. Ghamisi, and M. O. Ulfarsson, "Hyperspectral feature extraction using sparse and smooth low-rank analysis," *Remote Sens.*, vol. 11, no. 2, p. 121, 2019. doi: 10.3390/rs11020121. [Online]. Available: 10.3390/rs11020121

[44] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, p. 2319, 2000. doi: 10.1126/science.290.5500.2319.

[45] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 441–454, Mar. 2005. doi: 10.1109/TGRS.2004.842292.

[46] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. doi: 10.1126/science.290.5500.2323.

[47] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. 14th Int. Conf. Neural Information Processing Systems*, 2002, pp. 585–591.

[48] X. He and P. Niyogi, "Locality preserving projections," in *Proc. 16th Int. Conf. Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Scholkopf, Eds. Cambridge, MA: MIT Press, 2003, pp. 153–160.

[49] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan 2007. doi: 10.1109/TPAMI.2007.250598.

[50] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "Orsim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, 2019. doi: 10.1109/TGRS.2019.2897139.

[51] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, vol. 3. New York: Wiley, 1973.

[52] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion," *IEEE Trans. Geos. Remote Sens.*, vol. 54, no. 2, pp. 783–794, 2016. doi: 10.1109/TGRS.2015.2465899.

[53] C. Lee and D. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 388–400, 1993. doi: 10.1109/34.206958.

[54] B. Guo, S. R. Gunn, R. I. Damper, and J. D. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 522–526, 2006. doi: 10.1109/LGRS.2006.878240.

[55] H. Shi, Y. Shen, and Z. Liu, "Hyperspectral bands reduction based on rough sets and fuzzy c-means clustering," in *Proc. IEEE Instrumentation Technology Conf. (IMTC)*, 2003, vol. 2, pp. 1053–1056. doi: 10.1109/IMTC.2003.1207913.

[56] J. Yang and J. Yang, "Why can LDA be performed in PCA transformed space?" *Pattern Recognit.*, vol. 36, no. 2, pp. 563–566, 2003. doi:10.1016/S0031-3203(02)00048-1.

[57] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geos. Remote Sens.*, vol. 47, no. 3, pp. 862–873, 2009. doi: 10.1109/TGRS.2008.2005729.

[58] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1027–1061, May 2007.

[59] M. Imani and H. Ghassemian, "Feature space discriminant analysis for hyperspectral data feature reduction," *ISPRS J. Photogramm. Remote. Sens.*, vol. 102, pp. 1–13, Apr. 2015. doi: 10.1016/j.isprsjprs.2014.12.024.

[60] S. Patra, P. Modi, and L. Bruzzone, "Hyperspectral band selection based on rough set," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5495–5503, 2015. doi: 10.1109/TGRS.2015.2424236.

[61] X. Cao, T. Xiong, and L. Jiao, "Supervised band selection using local spatial information for hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 329–333, 2016. doi: 10.1109/LGRS.2015.2511186.

[62] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003. doi: 10.1162/089976603321780317.

[63] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Neural Information Processing Systems*, 2004, pp. 153–160.

[64] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Neural Information Processing Systems*, 2005, pp. 1601–1608.

[65] N. H. Ly, Q. Du, and J. E. Fowler, "Sparse graph-based discriminant analysis for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 3872–3884, 2014. doi: 10.1109/TGRS.2013.2277251.

[66] N. H. Ly, Q. Du, and J. E. Fowler, "Collaborative graph-based discriminant analysis for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2688–2696, 2014. doi: 10.1109/JSTARS.2014.2315786.

[67] D. Hong, N. Yokoya, and X. X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2960–2975, 2017. doi: 10.1109/JSTARS.2017.2682189.

[68] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," *ISPRS J. Photogramm. Remote. Sens.*, vol. 158, pp. 35–49, Dec. 2019. doi: 10.1016/j.isprsjprs.2019.09.008.

[69] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 471–478. doi: 10.1109/ICCV.2011.6126277.

[70] D. Hong and X. X. Zhu, "SULoRA: Subspace unmixing with low-rank attribute embedding for hyperspectral data analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1351–1363, 2018. doi: 10.1109/JSTSP.2018.2877497.

[71] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, 2019. doi: 10.1109/TIP.2018.2878958.

[72] H. Huang, F. Luo, J. Liu, and Y. Yang, "Dimensionality reduction of hyperspectral images based on sparse discriminant manifold embedding," *ISPRS J. Photogramm. Remote Sens.*, vol. 106, pp. 42–54, Aug. 2015. doi: 10.1016/j.isprsjprs.2015.04.015.

[73] Z. Xue, P. Du, J. Li, and H. Su, "Simultaneous sparse graph embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6114–6133, 2015. doi: 10.1109/TGRS.2015.2432059.

[74] W. Li and Q. Du, "Laplacian regularized collaborative graph for discriminant analysis of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7066–7076, 2016. doi: 10.1109/TGRS.2016.2594848.

[75] W. Li, J. Liu, and Q. Du, "Sparse and low-rank graph for discriminant analysis of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4094–4105, 2016. doi: 10.1109/TGRS.2016.2536685.

[76] L. Pan, H.-C. Li, Y.-J. Deng, F. Zhang, X.-D. Chen, and Q. Du, "Hyperspectral dimensionality reduction by tensor sparse and low-rank graph-based discriminant analysis," *Remote Sens.*, vol. 9, no. 5, p. 452, 2017. doi: 10.3390/rs9050452.

[77] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, 2001. doi: 10.1109/72.914517.

[78] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. European Conf. Computer Vision*, 2010, pp. 1–14. doi: 10.1007/978-3-642-15561-1_1.

[79] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. doi: 10.1561/2200000016.

[80] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999. doi: 10.1137/S0895479897326432.

[81] K.-C. Li, "Sliced inverse regression for dimension reduction," *J. Am. Stat. Assoc.*, vol. 86, no. 414, pp. 316–327, 1991. doi: 10.1080/01621459.1991.10475035.

[82] T. Suzuki and M. Sugiyama, "Sufficient dimension reduction via squared-loss mutual information estimation," *Neural Comput.*, vol. 25, no. 3, pp. 725–758, 2013. doi: 10.1162/NECO_a_00407.

[83] J. Sainui and M. Sugiyama, "Direct approximation of quadratic mutual information and its application to dependence-maximization clustering," *IEICE Trans. Inf. Syst.*, vol. 96, no. 10, pp. 2282–2285, 2013. doi: 10.1587/transinf.E96.D.2282.

[84] V. Tangkaratt, H. Sasaki, and M. Sugiyama, "Direct estimation of the derivative of quadratic mutual information with application in supervised dimension reduction," *Neural Comput.*, vol. 29, no. 8, pp. 2076–2122, 2017. doi: 10.1162/neco_a_00986.

[85] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "Cospace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geos. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, 2019. doi: 10.1109/TGRS.2018.2890705.

[86] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote. Sens.*, vol. 147, pp. 193–205, Jan. 2019. doi: 10.1016/j.isprsjprs.2018.10.006.

[87] D. Hong, N. Yokoya, J. Xu, and X. Zhu, "Joint & progressive learning from high-dimensional data for multi-label classification," in *Proc. European Conf. Computer Vision*, 2018, pp. 469–484. doi: 10.1007/978-3-030-01237-3_29.

[88] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, 2016. doi: 10.1109/MGRS.2016.2540798.

[89] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sept. 2019. doi: 10.1109/TGRS.2019.2907932.

[90] G. Licciardi and J. Chanussot, "Spectral transformation based on nonlinear principal component analysis for dimensionality reduction of hyperspectral images," *Eur. J. Remote Sens.*, vol. 51, no. 1, pp. 375–390, 2018. doi: 10.1080/22797254.2018.1441670.

[91] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014. doi: 10.1109/JSTARS.2014.2329330.

[92] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral

imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, 2015. doi: 10.1109/LGRS.2015.2482520.

[93] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2693–2705, 2017. doi: 10.1109/TGRS.2017.2651639.

[94] X. Kang, C. Li, S. Li, and H. Lin, "Classification of hyperspectral images by gabor filtering based deep network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1166–1178, 2018. doi: 10.1109/JSTARS.2017.2767185.

[95] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, 2019. doi: 10.1109/TGRS.2018.2868851.

[96] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, 2016. doi: 10.1109/JSTARS.2016.2517204.

[97] X. Sun, F. Zhou, J. Dong, F. Gao, Q. Mu, and X. Wang, "Encoding spectral and spatial context information for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2250–2254, 2017. doi: 10.1109/LGRS.2017.2759168.

[98] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 2250–2254, 2019. doi: 10.1109/TGRS.2019.2893180.

[99] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Sensors*, vol. 2015, pp. 1–12, July 2015. doi: 10.1155/2015/258619.

[100] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens*, vol. 54, no. 8, pp. 4544–4554, 2016. doi: 10.1109/TGRS.2016.2543748.

[101] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, 2016. doi: 10.1109/TGRS.2016.2584107.

[102] X. Liu, C. Deng, J. Chanussot, D. Hong, and B. Zhao, "StfNet: A two-stream convolutional neural network for spatiotemporal image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6552–6564, 2019. doi: 10.1109/TGRS.2019.2907310.

[103] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, 2018. doi: 10.1109/TGRS.2018.2794326.

[104] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, 2018. doi: 10.1109/TGRS.2018.2837142.

[105] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016. doi: 10.1109/LGRS.2016.2595108.

[106] Y. Zhan, D. Hu, H. Xing, and X. Yu, "Hyperspectral band selection based on deep convolutional neural network and distance density," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2365–2369, Dec. 2017. doi: 10.1109/LGRS.2017.2765339.

[107] Y. Kong, X. Wang, and Y. Cheng, "Spectral-spatial feature extraction for HSI classification based on supervised hypergraph and sample expanded CNN," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4128–4140, Nov 2018. doi: 10.1109/JSTARS.2018.2869210.

[108] J. Yang, Y. Zhao, and J. C. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Au. 2017. doi: 10.1109/TGRS.2017.2698503.

[109] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017. doi: 10.1109/TGRS.2017.2693346.

[110] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5585–5599, Oct. 2017. doi: 10.1109/TGRS.2017.2710079.

[111] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018. doi: 10.1109/TGRS.2018.2841823.

[112] M. Liang, L. Jiao, S. Yang, F. Liu, B. Hou, and H. Chen, "Deep multiscale spectral-spatial feature fusion for hyperspectral images classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2911–2924, Aug 2018. doi: 10.1109/JSTARS.2018.2836671.

[113] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, 2018. doi: 10.1109/TIP.2017.2772836.

[114] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 937–949, 2018. doi: 10.1109/TGRS.2017.2756851.

[115] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Trans. Geosci. Remote Sens*, vol. 56, no. 2, pp. 847–858, 2018. doi: 10.1109/TGRS.2017.2755542.

[116] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1330, 2017. doi: 10.3390/rs9121330.

[117] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMS," *Neurocomputing*, vol. 328, pp. 39–47, 2019. doi: 10.1016/j.neucom.2018.02.105.

[118] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, "Spatial sequential recurrent neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4141–4155, 2018. doi: 10.1109/JSTARS.2018.2844873.

[119] C. Shi and C.-M. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, no. 14, pp. 82–93, June 2018. doi: 10.1016/j.neucom.2018.03.012.

[120] A. Ma, A. M. Filippi, Z. Wang, and Z. Yin, "Hyperspectral image classification using similarity measurements-based deep recurrent neural networks," *Remote Sens.*, vol. 11, no. 2, pp. 194, 2019. doi: 10.3390/rs11020194.

[121] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, 2018. doi: 10.1109/TGRS.2018.2827407.

[122] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, 2019. doi: 10.1109/TGRS.2019.2899129.

[123] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, 2018. doi: 10.1109/TGRS.2017.2778343.

[124] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[125] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances Neural Information Processing Systems*, Lake Tahoe, NV, 2012, pp. 1527–1554. doi: 10.1145/3065386.

[126] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017. doi: 10.1109/MGRS.2017.2762087.

[127] B. Rasti, M. Ulfarsson, and J. Sveinsson, "Hyperspectral subspace identification using SURE," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2481–2485, Dec. 2015. doi: 10.1109/LGRS.2015.2485999.

[128] J. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geos. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008. doi: 10.1109/TGRS.2008.918089.

[129] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot, "Noise reduction in hyperspectral imagery: Overview and application," *Remote Sens.*, vol. 10, no. 3, p. 482, 2018. doi: 10.3390/rs10030482. [Online]. Available: http://www.mdpi.com/2072-4292/10/3/482

[130] IDC, "The digital universe of opportunities: Rich data and the increasing value of the Internet of Things," April 2014. [Online]. Available: https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

[131] BehnoodRasti, "HyFTech-Hyperspectral-Shallow-Deep-Feature-Extraction-Toolbox." Accessed on: Mar. 4, 2020. [Online]. Available: https://github.com/BehnoodRasti/HyFTech-Hyperspectral-Shallow-Deep-Feature-Extraction-Toolbox

[132] C. Debes et al., "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS Data Fusion Contest." *IEEE J. Sel. Topics Appl. Earth Observ. Remote Ses.*, vol. 7, no. 6, pp. 2405–2418, Mar. 2014. doi: 10.1109/JSTARS.2014.2305441.

[133] Y. Xu et al., "Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS Data Fusion Contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Ses.*, vol. 12, no. 6, pp. 1709–1724. doi: 10.1109/JSTARS.2019.2911113.

[134] "2018 IEEE GRSS Data Fusion Contest." [Online]. Available: http://www.grss-ieee.org/community/technical-committees/data-fusion

***GRS***