Folded LDA: Extending the Linear Discriminant Analysis Algorithm for Feature Extraction and Data Reduction in Hyperspectral Remote Sensing

Samson Damilola Fabiyi, *Student Member, IEEE*, Paul Murray, Jaime Zabalza, *Member, IEEE*, and Jinchang Ren, *Senior Member, IEEE*

***Abstract*—The rich spectral information provided by hyperspectral imaging has made this technology very useful in the classification of remotely sensed data. However, classification of hyperspectral data is typically affected by noise and the Hughes phenomenon due to the presence of hundreds of spectral bands and correlation among them, with usually a limited number of samples for training. Linear discriminant analysis (LDA) is a well-known technique that has been widely used for supervised dimensionality reduction of hyperspectral data. However, the use of LDA in hyperspectral remote sensing is limited due to its poor performance on small training datasets and the limited number of features that can be selected i.e., $c-1$, where c is the number of classes in the data. To solve these problems, this article presents a folded LDA (F-LDA) for dimensionality reduction of remotely sensed HSI data in small sample size scenarios. The proposed approach allows many more discriminant features to be selected in comparison to the conventional LDA since the selection is no longer bound by the limiting factor, leading to significantly higher accuracy in the classification of pixels under SSS restrictions. The proposed approach is evaluated on five different datasets, where the experimental results demonstrate the superiority of the F-LDA to the conventional LDA in terms of not only higher classification accuracy but also reduced computational complexity, and reduced contiguous memory requirements.**

***Index Terms*—Dimensionality reduction, folded linear discriminant analysis (F-LDA), hyperspectral remote sensing, small sample size (SSS) scenario, supervised feature extraction.**

I. Introduction

THE past few years have witnessed the availability of images with very high spectral resolution through the development of Hyperspectral Imaging (HSI) sensors [1]. The rich spectral information [2], [3] provided by these images has made them very useful in remote sensing applications such as classification and target detection. Recently, classification of hyperspectral remotely sensed images has received more attention due to their major roles in precision agriculture [4], environment monitoring [5], and national security [6], among others.

However, hyperspectral data classification is faced with the problem of small sample size (SSS) scenario (limited number of labeled samples for training) and the presence of many spectral bands, usually in order of hundreds [7]–[9]. In addition, many of these bands are highly correlated resulting in data redundancy and noise [10], [11]. As a result of this lack of sufficient samples for training and the high dimensionality of HSI data, the performance of traditional machine learning classifiers in HSI is often affected by the Hughes phenomenon [12]–[14]. The Hughes phenomenon is characterized by a decline in the classification accuracy after it reaches a maximum following an initial rise as more spectral bands are added [15]–[17]. The Hughes phenomenon therefore limits the classifying ability of conventional classification models leading to reduced accuracy and increased computational complexity. Hence, the dimensions of HSI data are often reduced through feature extraction techniques before they are presented to the models for classification [18]. Applying these techniques to transform the hyperspectral data into a lower dimensional space is therefore capable of not only increasing the classification accuracy but also reducing the computational complexity and memory requirement.

Unlike deep learning-based classifiers, which require complex parameter settings and involve feature extraction from the raw hyperspectral data (feature extraction and classification come in the same package) [12], [19], the traditional machine learning classifiers require simple parameter settings and are inputted the outputs of separate feature extraction techniques [19]. These techniques can be categorized into supervised and unsupervised approaches depending on whether the class labels are included in the feature extraction process or not. Examples of supervised feature extraction techniques are linear discriminant analysis (LDA) [20], generalized discriminant analysis (GDA) [21], and nonparametric weighted feature extraction (NWFE) [22], while techniques, such as principal component analysis (PCA) [23], singular spectral analysis (SSA) [24], independent component analysis (ICA) [25], singular value decomposition (SVD) [26], locally linear embedding [27], isometric mapping [28], random projection [29], maximum noise fraction [30] and wavelet dimensionality reduction [31] are unsupervised (they

Manuscript received August 13, 2021; revised October 20, 2021 and November 5, 2021; accepted November 12, 2021. Date of publication November 23, 2021; date of current version December 13, 2021. This work was supported by the University of Strathclyde, Glasgow, U.K. *(Corresponding author: Samson Damilola Fabiyi.)*

Samson Damilola Fabiyi, Paul Murray, and Jaime Zabalza are with the Department of Electronic and Electrical Engineering, University of Strathclyde, G11XW Glasgow, U.K. (e-mail: samson.fabiyi@strath.ac.uk; paul.murray@strath.ac.uk; j.zabalza@strath.ac.uk).

Jinchang Ren is with the National Subsea Centre, Robert Gordon University, AB10 7AQ Aberdeen, U.K. (e-mail: j.ren@rgu.ac.uk; jinchang.ren@ieee.org).

Digital Object Identifier 10.1109/JSTARS.2021.3129818

do not require labeled data to extract the features). While a large part of the current state of the art is focusing on deep learning-based approaches, significant attention is still being directed to those more traditional methodologies for improved feature extraction leading to better classification results. For instance, a non-linear version of the PCA, known as Kernel PCA (KPCA), was implemented in [32] where the HSI data was transformed into a linearly separable feature space to capture higher order statistics and extract the principal components for classification. Also, in [33], the performance of the SSA was optimized by applying SVD on a representative pixel as opposed to every pixel in the hyperspectral data resulting in similar classification results but reduced computational complexity. In [34], the PCA was extended by folding each of the spectral vectors in the hyperspectral data matrix, unfolding the projected samples for classification. The increased accuracy and reduced computational complexity achieved in [34] motivate us to extend LDA using a similar mathematical trick.

LDA is a feature extraction technique that has found many recent applications in HSI data classification [35], [36]. LDA, as a supervised data reduction technique, aims to extract features which maximize the separability among the different classes in the data. LDA achieves this by using a transformation matrix to project the original data unto a lower dimensional space. The transformation matrix is computed to maximize the between-class variance and minimize the within-class variance. The size of the between-class matrix, the within-class matrix and the resulting transformation matrix is dependent on the dimensionality of the data. Specifically, their size is given as $f * f$ where f is the number of features (spectral bands) in the data.

It is well known that in LDA, the rank of the between-class variance matrix imposes a limitation on the number of components that can be extracted to $c-1$ where c is the number of classes in the data [37]–[39]. Also, due to the high dimensional nature of the HSI data, the different stages of the LDA require the processing and storage of very large matrices resulting in the problem of large memory requirement and high computational cost [40]–[42]. Finally, the traditional LDA as a dimensionality reduction technique does not perform well when the number of samples used to train the model is small as demonstrated in related work [18], [20], [43]–[45]. The use of the traditional LDA is therefore limited in hyperspectral remote sensing since enough labeled data is usually not available for training.

In [46]–[48], 2-D LDA was used in an application involving face recognition to solve the problem of SSS scenario. In a more related work [49], 2-D LDA was applied to hyperspectral data in where each of the pixels was transformed into feature matrix to reduce the effect of the SSS on the classification results. However, the concept of "folding the pixel," introduced for PCA in [34], was not considered, hence unfolding of the projected samples was also not included. Instead, the authors combined the eigenvectors into a single projection vector using a weighted sum. This continues to limit the number of features to be extracted to the number of columns in the feature matrices (folded pixels). Also, the performance of the 2-D LDA features was not benchmarked against the original feature space. Hence, it is not fully clear how effective the 2-D LDA can be in SSS scenarios since the goal of the feature extraction and data reduction is to improve the performance of the classifier on the original feature space. Furthermore, computational complexity analysis and experiments to show the limit to which this folding concept can be applied were not conducted.

Inspired by the folded PCA concept proposed in [34], this article presents an improved version of the LDA transform named folded-LDA (F-LDA), which shares some concepts with the 2-D LDA but improves it, further developing and analyzing the folding context. In the F-LDA, each pixel in the hyperspectral data is folded from vector to matrix. Different matrices sizes (configurations) were exploited and extensive experiments conducted to explain the folding limits, particularly when the dimension of the converted matrix is either $f * 1$ or $1 * f$. Eigenvectors were dealt with individually and the final features extracted following the unfolding of the projected samples. Consequently, the number of extracted features is no longer given as c-1 but the product of the number of columns in the converted matrices (folded pixels) and the rank of the between-class variance matrix. This gives room for the selection of many more discriminant features thereby making the proposed approach more flexible than the conventional LDA, also leading to more informative features (capturing local structure of the data thanks to the folded samples), and targeting higher classification accuracy than LDA, 2-D LDA and the original feature space. Computational complexity analysis is also conducted to illustrate additional benefits of the proposed approach which are summarized as follows.

1) The proposed approach requires less computational complexity to calculate the within-class variance, between-class variance, transformation matrix, and eigenvectors. Also, the cost of projecting the hyperspectral data into lower dimensional space is reduced.
2) The contiguous memory required for the proposed approach is much less than what is required for the conventional LDA.

The rest of this article is organized as follows. Sections II and III describe the related background and the proposed method, respectively. Description of data and experimental settings are presented in Section IV. Experimental results and performance evaluation of the proposed approach are presented in Section V. Finally, Section VI concludes this article.

II. Relevant Background: Conventional LDA for HSI Data

The conventional LDA is implemented in the following three steps: within-class variance and between-class variance computation; transformation matrix and eigenvectors computation; and data projection. The following sections present the full description of the traditional LDA implementation steps and applications for feature extraction and data reduction of HSI data.

A. Within-Class and Between-Class Variance Computation

The hyperspectral images are data cubes which consist of a set of 2-D images I (of rows k and columns l) captured at different wavelengths of the acquiring HSI sensors. Hence,

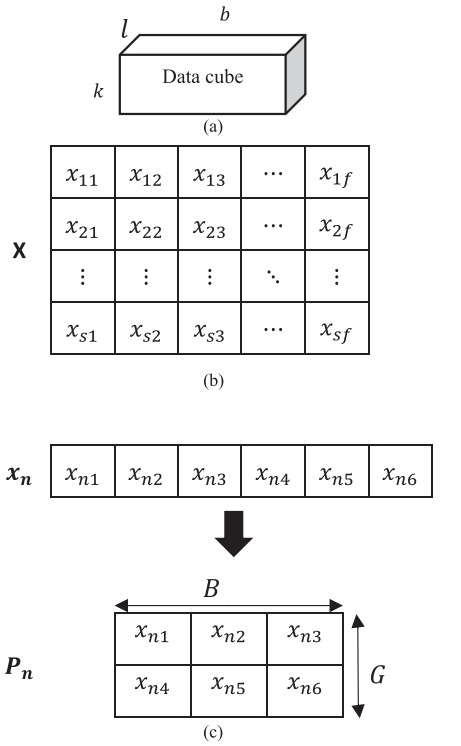


Fig. 1. (a) Hyperspectral data. (b) Data matrix X where each row depicts a spectra vector (sample), $x_n = [x_{n1}\ x_{n2}\ x_{n3} \ldots\ldots\ldots\ x_{nf}]$. (c) Spectral vector x_n is folded to form a 2-D matrix, P_n where $n \in [1,\ s]$, $G = 2$, $B = 3$ and $f = G \times B = 6$.

TABLE I
ALGORITHMIC STEP CODE OF THE PROPOSED F-LDA

Step	Algorithmic code
1.	Convert the hyperspectral data cube to a data matrix $\boldsymbol{X}$
2.	Fold each spectral vector $\boldsymbol{x_n}$ in the data matrix to a 2D matrix $\boldsymbol{P_n}$ to form a set of 2D matrices
3.	Compute the mean $\boldsymbol{M_j}$ of all $\boldsymbol{P_n}$ (folded samples) in each class
4.	Compute the mean $\boldsymbol{M}$ of all $\boldsymbol{P_n}$ (folded samples) in the data
5.	Use matrices $\boldsymbol{M}$ and $\boldsymbol{M_j}$ to compute the within-class variance matrix $\boldsymbol{V_{PW}}$ and between-class variance matrix $\boldsymbol{V_{PB}}$
6.	Compute the transformation matrix $\boldsymbol{T_P}$ using (13)
7.	Compute the eigenvectors and eigenvalues of $\boldsymbol{T_P}$
8.	Rank the eigenvectors in descending order according to their eigenvalues
9.	Use the first k eigenvectors to project the data into a lower dimensional space as in (14)
10.	Unfold the projected matrices

each pixel in the data cube is a set of bands of reflected light in the range of the wavelengths of the sensors [50]–[52]. If the dimension of the data cube is $k \times l \times b$, then b is the number of spectral bands [53].

For linear discriminant analysis, the data cube, as shown in Fig. 1, is converted into a data matrix X of s rows and f columns where $s = k \times l$ and $f = b$ are the number of samples and features, respectively. Each sample in X, denoted by x_n where $n \in [1,\ s]$, is a spectral vector of a pixel in the data cube. Let the number of classes and jth class in X be c and c_j respectively, the number of samples in each class can be denoted as N_j and the ith sample in class c_j denoted as x_{ij} where $i \in [1,\ N_j]$. The mean of the spectral vectors in each class c_j, denoted as m_j where $j \in [1,\ c]$, and the overall mean of the data matrix X, denoted as m, are then calculated using

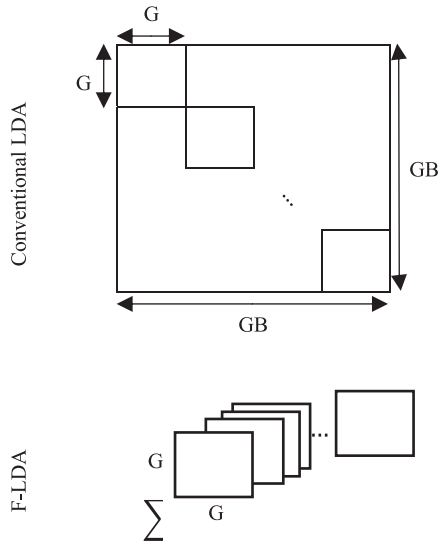


Fig. 2. Comparing the within-class variance matrices constructed using the F-LDA and the conventional LDA. The F-LDA matrix is based on the variance matrices constructed using the accumulation of those blocks across the main diagonal of the conventional LDA matrix, leading to a local extraction of features.

$$m_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_{ij} \tag{1}$$

$$m = \sum_{j=1}^{c} \frac{N_j}{s} m_j. \tag{2}$$

The within-class variance V_W and the between-class variance V_B of X are computed using

$$V_W = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_{ij} - m_j)^T (x_{ij} - m_j) \tag{3}$$

$$V_B = \sum_{j=1}^{c} N_j (m_j - m)^T (m_j - m) \tag{4}$$

where $V_W \in \Re^{f \times f}$ and $V_B \in \Re^{f \times f}$.

B. Transformation Matrix, Eigenvectors Computation, and Data Projection

The transformation matrix, T, is computed to maximize the between-class variance, V_B, and minimize the within-class variance, V_W, using

$$T = {V_W}^{-1} V_B,\ T \in \Re^{f \times f}. \tag{5}$$

TABLE II
CLASSIFICATION ACCURACY (%) RESULTS (BEST CASES) FOR THE BOTSWANA DATASET (14 CLASSES) USING ORIGINAL FEATURE SPACE, CONVENTIONAL LDA, F-LDA (WITH DIFFERENT CONFIGURATIONS), 2-D LDA, GDA, NWFE, KPCA AND F-PCA

Sample Shape	V_B Matrix Rank (r)/ EVD Components (d_{EVD})	Best d_{EVD}	Number of Features (d_{TOTAL})	OA (%)	AA (%)	k (%)
Original Feature Space						
1 × 145	N/A	N/A	145	87.60 ± 0.73	88.54 ± 0.75	86.56 ± 0.01
Conventional LDA						
1 × 145	13	12	12	13.31 ± 5.14	13.04 ± 2.68	6.07 ± 0.03
Folded-LDA						
1 × 145	N/A	N/A	145	87.60 ± 0.73	88.54 ± 0.75	86.56 ± 0.01
5 × 29	5	5	145	89.79 ± 1.03	90.68 ± 1.08	88.94 ± 0.01
29 × 5	**29**	**12**	**60**	**91.17 ± 1.06**	**91.69 ± 1.15**	**90.43 ± 0.01**
145 × 1	13	12	12	13.31 ± 5.14	13.04 ± 2.68	6.07 ± 0.03
2D-LDA						
1 × 145	N/A	N/A	145	87.60 ± 0.73	88.54 ± 0.75	86.56 ± 0.01
5 × 29	5	N/A	29	79.46 ± 2.59	80.50 ± 2.91	77.75 ± 0.03
29 × 5	29	N/A	5	69.81 ± 5.59	70.67 ± 5.42	67.29 ± 0.06
145 × 1	13	N/A	1	14.15 ± 3.37	14.40 ± 3.38	7.44 ± 0.04
GDA						
1 × 145	13	13	13	85.72 ± 1.04	85.97 ± 1.85	84.52 ± 0.01
NWFE						
1 × 145	20	5	5	87.57 ± 1.77	88.10 ± 2.29	86.53 ± 0.02
KPCA						
1 × 145	10	7	7	89.97 ± 1.15	90.41 ± 1.51	89.13 ± 0.01
Folded PCA						
29 × 5	5	1	29	87.47 ± 1.07	88.47 ± 1.05	86.43 ± 0.01

The eigenvalues λ and the eigenvectors v of T are computed. The eigenvectors v are then ranked, starting from the highest to the lowest, using their corresponding eigenvalues. The ranked eigenvector v with a dimension of $f * f$ is later reduced to V_d of f rows and d columns by selecting the first d columns of the ranked v and discarding the remaining eigenvectors (with small eigenvalues). It is worth stating here that the value of d is bound by the number of non-zero eigenvalues, which is $c - 1$, and is also given as the rank of the between-class variance matrix, V_B [37]–[39].

The data matrix X is then projected unto a lower dimensional space Y using

$$Y = XV_d,\ X \in \Re^{s \times f},\ V_d \in \Re^{f \times d} \tag{6}$$

where Y and X are the projected data and the original data, respectively.

III. PROPOSED FOLDED-LDA

A. Concepts of the Proposed F-LDA

In our proposed F-LDA, each spectral vector (sample) in X is folded into a matrix, as shown in Fig. 1. This is the main difference from the conventional LDA, which treats the pixels as spectral vectors. By folding the spectral vectors into matrices, our approach can generate alternative variance matrices for capturing information in a new and different way, where the content across contiguous spectral bands (local structures) is thus highlighted [33], [34].

In the proposed F-LDA, after converting all the spectral vectors in the data matrix into matrices of the same configuration (different sizes of the matrix will be exploited as will be discussed in Section III-D), the data in a class becomes a stack of matrices (folded samples) belonging to that class. With these, the mean of each class and the overall mean of the data are computed as matrices, from which the within-class variance, between-class variance and the transformation matrix are also computed.

The folding of each spectral vector, whose length is the number of features in the data matrix, is done in such a way that the folded samples are of size $G \times B$, where G is the number of groups and B is number of bands in each group, which is common to all groups for simplicity. As will be shown later in Section V, $G = f$ is a special case of our F-LDA in which the number of groups in the folded sample is f and F-LDA simplifies to the conventional LDA. Similarly, $G = 1$ is another special case where the number of groups in the folded sample is 1 and F-LDA simplifies to original data matrix. The final step in the F-LDA is the unfolding of the projected samples which are presented to classification models for discrimination.

The transformation matrix obtained using the stack of folded samples provides an alternative but more effective way to extract features, which are now many more than $c - 1$, from the HSI

TABLE III
CLASSIFICATION ACCURACY (%) RESULTS (BEST CASES) FOR THE PAVIA CENTER DATASET (NINE CLASSES) USING ORIGINAL FEATURE SPACE, CONVENTIONAL LDA, F-LDA (WITH DIFFERENT CONFIGURATIONS), 2-D LDA, GDA, NWFE, KPCA, AND F-PCA

Sample Shape	V_B Matrix Rank (r)/ EVD Components (d_{EVD})	Best d_{EVD}	Number of Features (d_{TOTAL})	OA (%)	AA (%)	k (%)
Original Feature Space						
1 × 102	N/A	N/A	102	95.44 ± 0.53	84.85 ± 1.51	93.54 ± 0.01
Conventional LDA						
1 × 102	8	6	6	90.76 ± 1.63	76.50 ± 3.33	86.92 ± 0.02
Folded-LDA ($Y_n = P_n{}^T V_{Pd}$)						
1×102	N/A	N/A	102	95.44 ± 0.53	84.85 ± 1.51	93.54 ± 0.01
2×51	2	2	102	95.23 ± 0.65	84.69 ± 1.90	93.24 ± 0.01
3×34	3	3	102	95.69 ± 0.56	84.69 ± 3.04	93.90 ± 0.01
6×17	6	6	102	96.63 ± 0.39	87.76 ± 2.11	95.23 ± 0.01
17×6	**17**	**9**	**54**	**96.63 ± 0.37**	**88.06 ± 1.51**	**95.23 ± 0.01**
34×3	24	8	24	96.21 ± 0.26	87.25 ± 0.85	94.63 ± 0.00
51×2	16	8	16	95.28 ± 0.53	84.69 ± 1.54	93.30 ± 0.01
102×1	8	6	6	90.76 ± 1.63	76.50 ± 3.33	86.92 ± 0.02
2D-LDA						
1×102	N/A	N/A	102	95.44 ± 0.53	84.85 ± 1.51	93.54 ± 0.01
2×51	2	N/A	51	93.49 ± 1.45	81.31 ± 3.20	90.74 ± 0.02
3×34	3	N/A	34	92.13 ± 0.68	74.40 ± 2.72	88.77 ± 0.01
6×17	6	N/A	17	88.90 ± 4.23	66.30 ± 10.03	83.90 ± 0.06
17×6	17	N/A	6	86.27 ± 3.27	61.88 ± 8.42	80.22 ± 0.05
34×3	24	N/A	3	77.92 ± 7.11	47.60 ± 9.42	66.89 ± 0.12
51×2	16	N/A	2	75.64 ± 7.91	44.38 ± 8.65	63.51 ± 0.13
102×1	8	N/A	1	66.20 ± 8.87	37.62 ± 5.40	52.00 ± 0.13
GDA						
1 × 102	8	8	8	95.73 ± 0.42	85.35 ± 1.93	93.95 ± 0.01
NWFE						
1 × 102	20	4	4	94.73 ± 0.58	81.71 ± 2.98	92.52 ± 0.01
KPCA						
1 × 102	10	10	10	95.49 ± 0.51	84.60 ± 2.13	93.61 ± 0.01
Folded PCA						
3×34	5	3	9	96.17 ± 0.51	86.87 ± 2.18	94.58 ± 0.01

data which in turn brings about improved classification results, less computational complexity and reduced contiguous memory requirement. It is worth noting that this improvement in the classification results is dependent on the configuration of the folded samples, $G \times B$.

B. Implementation of the Proposed F-LDA

As shown in Fig. 1, denoting a spectral vector in the data matrix as $x_n = [x_{n1}\ x_{n2}\ x_{n3} \ldots\ldots\ldots\ x_{nf}]$ where $n \in [1,\ s]$, a folded sample (resulting matrix), P_n, of this vector can be denoted using

$$P_n = \begin{bmatrix} p_{n(11)} & \cdots & p_{n(1,B))} \\ \vdots & \ddots & \vdots \\ p_{n(G,1)} & \cdots & p_{n(G,B)} \end{bmatrix} \tag{7}$$

and each element in the matrix P_n, is denoted as $p_{n(h+1,\,i)}$ and computed using

$$p_{n(h+1,\,i)} = x_{(h*B)+i} \tag{8}$$

where $h \in [0,\ G-1]$ and $i \in [1,\ B]$.

The mean of the converted matrices in each class c_j, denoted as M_j where $j \in [1,\ c]$, and the overall mean of all the converted matrices, denoted as M, are then calculated using

$$M_j = \frac{1}{N_j} \sum_{i=1}^{N_j} P_{ij}\ ,\ M_j \in \Re^{G \times B} \tag{9}$$

$$M = \sum_{j=1}^{c} \frac{N_j}{s} M_j,\ M \in \Re^{G \times B} \tag{10}$$

where P_{ij} is the ith converted matrix in class c_j and $i \in [1,\ N_j]$.

The within-class variance V_{PW} and the between-class variance V_{PB} of the data cube are computed using

$$V_{PW} = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (P_{ij} - M_j)(P_{ij} - M_j)^T \tag{11}$$

$$V_{PB} = \sum_{j=1}^{c} N_j (\mathbf{M}_j - \mathbf{M})(\mathbf{M}_j - \mathbf{M})^T \tag{12}$$

where $V_{PW} \in \Re^{G \times G}$ and $V_{PB} \in \Re^{G \times G}$.

TABLE IV
CLASSIFICATION ACCURACY (%) RESULTS (BEST CASES) FOR THE SALINAS DATASET (16 CLASSES) USING ORIGINAL FEATURE SPACE, CONVENTIONAL LDA, F-LDA (WITH DIFFERENT CONFIGURATIONS), 2-D LDA, GDA, NWFE, KPCA AND F-PCA

Sample Shape	V_B Matrix Rank (r)/ EVD Components (d_{EVD})	Best d_{EVD}	Number of Features (d_{TOTAL})	OA (%)	AA (%)	k (%)
Original Feature Space						
1 × 204	N/A	N/A	204	87.20 ± 0.93	90.82 ± 1.18	85.71 ± 0.01
Conventional LDA						
1 × 204	15	13	13	54.71 ± 12.82	48.00 ± 15.67	47.39 ± 0.18
Folded-LDA ($Y_n = P_n{}^T V_{Pd}$)						
1 × 204	N/A	N/A	204	87.20 ± 0.93	90.82 ± 1.18	85.71 ± 0.01
2 × 102	2	2	204	88.07 ± 0.78	91.26 ± 1.21	86.68 ± 0.01
3 × 68	3	3	204	88.23 ± 0.78	91.38 ± 0.99	86.85 ± 0.01
4 × 51	4	4	204	88.47 ± 0.54	91.86 ± 0.88	87.13 ± 0.01
6 × 34	6	5	170	89.21 ± 0.43	92.81 ± 0.67	87.96 ± 0.00
12 × 17	12	8	136	89.48 ± 0.69	93.18 ± 0.99	88.27 ± 0.01
17 × 12	**17**	**10**	**120**	**90.06 ± 0.76**	**94.10 ± 0.65**	**88.91 ± 0.01**
204 × 1	15	13	13	54.71 ± 12.82	48.00 ± 15.67	47.39 ± 0.18
2D-LDA						
1 × 204	N/A	N/A	204	87.20 ± 0.93	90.82 ± 1.18	85.71 ± 0.01
2 × 102	2	N/A	102	85.30 ± 1.86	88.25 ± 1.88	83.58 ± 0.02
3 × 68	3	N/A	68	86.51 ± 1.53	89.29 ± 1.73	84.93 ± 0.02
4 × 51	4	N/A	51	86.00 ± 0.97	88.77 ± 1.93	84.35 ± 0.01
6 × 34	6	N/A	34	84.67 ± 1.41	87.82 ± 1.91	82.88 ± 0.02
12 × 17	12	N/A	17	84.81 ± 1.26	86.91 ± 2.24	83.03 ± 0.01
17 × 12	17	N/A	12	83.29 ± 1.69	85.78 ± 3.19	81.34 ± 0.02
204 × 1	15	N/A	1	25.98 ± 3.99	25.49 ± 4.56	19.54 ± 0.05
GDA						
1 × 204	15	14	14	88.44 ± 1.35	93.17 ± 0.72	87.11 ± 0.01
NWFE						
1 × 204	20	7	7	84.38 ± 1.23	87.81 ± 1.50	82.56 ± 0.01
KPCA						
1 × 204	10	10	10	88.82 ± 0.95	92.09 ± 1.21	87.52 ± 0.01
Folded PCA						
6 × 34	5	5	30	88.95 ± 1.30	92.71 ± 0.81	87.66 ± 0.01

From the between-class variance V_{PB} and the within-class variance V_{PW} the same approach in Section II-B is applied to compute the transformation matrix T_P the eigenvalues λ_P, the eigenvectors V_P and the selected eigenvectors V_{Pd}. The data are then projected unto a lower dimensional space using (14) where Y_n is the projected matrix of each sample

$$T_P = {V_{PW}}^{-1} V_{PB}\,,\ T_P \in \Re^{G \times G} \tag{13}$$

$$Y_n = {P_n}^T\, V_{Pd},\ V_{Pd} \in \Re^{G \times d'},\ Y_n \in \Re^{G \times d'}. \tag{14}$$

Finally, the algorithmic step code for the proposed approach is given in Table I. The dimensions of the between-class variance V_{PB}, and the within-class variance V_{PW}, computed using the F-LDA are $G \times G$ while the dimensions of the between-class variance V_B and the within-class variance V_W computed using the conventional LDA is $f \times f$ or $GB \times GB$. Similarly, the dimensions of the transformation matrix T_P are $G \times G$ while the dimension of the transformation matrix from the conventional LDA are $f \times f$ or $GB \times GB$. The computational complexity of calculating the within-class variance, between-class variance, transformation matrix, eigenvalues and eigenvectors is therefore reduced significantly as will be shown in Section V. The projected data are obtained by the multiplication of ${P_n}^T$ and V_{Pd}. These are two smaller matrices of size $B \times G$ and $G \times d$ respectively and contribute to the significant reduction in the computational complexity.

C. Extraction of Local Structures Using the Proposed Approach

This section explains how the proposed approach can capture the local structure in the spectral vectors. Denoting each row in each of the folded matrices $P_{ij} = P_n$ in (6) as p_{ijk} where

TABLE V
CLASSIFICATION ACCURACY (%) RESULTS (BEST CASES) FOR THE INDIAN PINE DATASET (16 CLASSES) USING ORIGINAL FEATURE SPACE, CONVENTIONAL LDA, F-LDA (WITH DIFFERENT CONFIGURATIONS), 2-D LDA, GDA, NWFE, KPCA, AND F-PCA

Sample Shape	V_B Matrix Rank (r)/ EVD Components (d_{EVD})	Best d_{EVD}	Number of Features (d_{TOTAL})	OA (%)	AA (%)	k (%)
Original Feature Space						
1 × 200	N/A	N/A	200	64.73 ± 1.87	76.53 ± 1.38	60.51 ± 0.02
Conventional LDA						
1 × 200	15	3	3	22.95 ± 3.59	31.55 ± 6.65	16.65 ± 0.03
Folded-LDA ($Y_n = P_n^T V_{Pd}$)						
1×200	N/A	N/A	200	64.73 ± 1.87	76.53 ± 1.38	60.51 ± 0.02
2×100	2	2	200	69.92 ± 1.46	79.86 ± 0.98	66.22 ± 0.02
4×50	4	3	150	72.40 ± 1.90	82.16 ± 1.06	68.96 ± 0.02
5×40	5	4	160	70.91 ± 1.87	81.71 ± 1.02	67.33 ± 0.02
8×25	8	7	175	73.50 ± 1.75	83.18 ± 1.16	70.23 ± 0.02
10×20	10	6	120	72.92 ± 2.02	83.23 ± 0.76	69.55 ± 0.02
20×10	**20**	**13**	**130**	**73.99 ± 1.93**	**83.96 ± 1.25**	**70.75 ± 0.02**
200×1	15	3	3	22.95 ± 3.59	31.55 ± 6.65	16.65 ± 0.03
2D-LDA						
1×200	N/A	N/A	200	64.73 ± 1.87	76.53 ± 1.38	60.51 ± 0.02
2×100	2	N/A	100	62.34 ± 2.92	74.69 ± 2.78	57.88 ± 0.03
4×50	4	N/A	50	61.38 ± 2.44	72.45 ± 3.10	56.78 ± 0.03
5×40	5	N/A	40	53.99 ± 4.92	65.74 ± 4.72	48.70 ± 0.05
8×25	8	N/A	25	56.31 ± 3.33	68.37 ± 2.55	51.25 ± 0.04
10×20	10	N/A	20	54.79 ± 3.21	66.39 ± 3.49	49.52 ± 0.04
20×10	20	N/A	10	47.22 ± 3.24	60.49 ± 2.31	41.36 ± 0.03
200×1	15	N/A	1	14.35 ± 3.92	16.96 ± 2.75	7.93 ± 0.03
GDA						
1×200	15	15	15	61.57 ± 1.26	72.99 ± 0.94	57.02 ± 0.01
NWFE						
1×200	20	10	10	70.61 ± 2.65	80.66 ± 1.42	67.02 ± 0.03
KPCA						
1×200	10	7	7	65.03 ± 1.52	76.34 ± 1.65	60.85 ± 0.02
Folded PCA						
8×25	5	4	32	72.43 ± 1.51	82.52 ± 1.22	69.00 ± 0.02

$k \in [1,\ G]$, P_{ij} can be expressed as

$$P_{ij} = \begin{bmatrix} p_{ij1} \\ p_{ij2} \\ \vdots \\ p_{ijG} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_G \end{bmatrix}. \tag{15}$$

Each spectral vector in the data matrix can then be represented as $x_{ij} = x_n = [\, p_{n1}\ p_{n2}\ \ldots\ p_{nG}\,]$. If m_j, the overall mean of each class in the conventional LDA, is folded into a $G \times B$ matrix, m_j can also be expressed using

$$M_{mJ} = \begin{bmatrix} m_{j1} \\ m_{J2} \\ \vdots \\ m_{JG} \end{bmatrix}. \tag{16}$$

The within-class variance V_W in (3) as used in the conventional LDA can then be formulated using (17), shown at the bottom of the next page,

Also, let each row in M_j be denoted as t_{ji} where $i \in [1,\ G]$, M_J can be expressed as

$$M_J = \begin{bmatrix} t_{j1} \\ t_{J2} \\ \vdots \\ t_{JG} \end{bmatrix} = \begin{bmatrix} m_{j1} \\ m_{J2} \\ \vdots \\ m_{JG} \end{bmatrix} = M_{mJ}. \tag{18}$$

Finally, the within-class variance V_{PW} in (11) as used in the proposed approach can then be formulated using

$$V_{PW} = \sum_{j=1}^{c} \sum_{i=1}^{N_j} \Big[(p_1 - t_{j1})(p_1 - t_{j1}) + (p_2 - t_{j2})(p_2 - t_{j2}) + \cdots + (p_G - t_{jG})(p_G - t_{jG}) \Big]. \tag{19}$$

TABLE VI
CLASSIFICATION ACCURACY (%) RESULTS (BEST CASES) FOR THE PAVIA UNIVERSITY DATASET (9 CLASSES) USING ORIGINAL FEATURE SPACE, CONVENTIONAL LDA, F-LDA (WITH DIFFERENT CONFIGURATIONS), 2D LDA, GDA, NWFE, KPCA, AND F-PCA

Sample Shape	V_B Matrix Rank (r)/ EVD Components (d_{EVD})	Best d_{EVD}	Number of Features (d_{TOTAL})	OA (%)	AA (%)	k (%)
Original Feature Space						
1 × 103	N/A	N/A	103	85.02 ± 1.47	79.61 ± 3.20	79.84 ± 0.02
Conventional LDA						
1 × 103	8	5	5	66.45 ± 1.66	57.07 ± 3.80	55.86 ± 0.02
Folded-LDA ($Y_n = P_n{}^T V_{Pd}$)						
1 × 103	N/A	N/A	103	85.02 ± 1.47	79.61 ± 3.20	79.84 ± 0.02
2 × 52	2	2	104	84.77 ± 1.30	79.85 ± 2.61	79.49 ± 0.02
4 × 26	4	4	104	84.21 ± 0.98	79.30 ± 3.06	78.74 ± 0.01
8 × 13	**8**	**6**	**78**	**86.43 ± 1.16**	**81.05 ± 2.42**	**81.77 ± 0.02**
13 × 8	13	10	80	85.06 ± 1.49	79.14 ± 3.20	79.95 ± 0.02
103 × 1	8	5	5	66.45 ± 1.66	57.07 ± 3.80	55.86 ± 0.02
2D-LDA						
1 × 103	N/A	N/A	103	85.02 ± 1.47	79.61 ± 3.20	79.84 ± 0.02
2 × 52	2	N/A	52	76.52 ± 1.26	66.61 ± 3.44	67.41 ± 0.02
4 × 26	4	N/A	26	69.76 ± 2.99	56.12 ± 6.71	57.72 ± 0.05
8 × 13	8	N/A	13	74.42 ± 1.80	64.31 ± 5.96	64.76 ± 0.03
13 × 8	13	N/A	8	69.95 ± 3.07	58.99 ± 5.57	58.65 ± 0.04
103 × 1	8	N/A	1	51.49 ± 6.01	33.80 ± 5.85	32.30 ± 0.11
GDA						
1 × 103	8	7	7	77.76 ± 1.93	70.91 ± 2.30	70.28 ± 0.02
NWFE						
1 × 103	20	17	17	80.32 ± 1.29	70.21 ± 2.42	73.28 ± 0.02
KPCA						
1 × 103	10	9	9	80.33 ± 2.19	70.96 ± 3.77	73.22 ± 0.02
Folded PCA						
13 × 8	5	3	39	**83.86 ± 0.99**	76.92 ± 4.99	78.19 ± 0.01

Diagonal elements of V_W in (17) are accumulated to construct the within-class variance matrix, V_{PW} in (11) as shown in (19) and illustrated in Fig. 2. The local structures within the group bands are therefore covered and features that improve discrimination are extracted.

D. Different Configurations and Their Implications

In the proposed F-LDA, different configurations ($G \times B$) of the matrices can be exploited for all the data utilized in this article. The configurations were selected using the factors of f, the number of features in the data matrix. The total number of features extracted after applying the F-LDA in each case is d and is given as $B \times d_{\mathrm{EVD}}$, where d_{EVD} is the number of extracted components at the eigenvalue decomposition (EVD) of the transformation matrix, T_P. The number of discriminant components extracted is bound by the number of nonzero eigenvalues, which is given as the rank of the between-class variance matrix, V_{PB} [37]–[39]. It is therefore required that the value of d_{EVD} be varied from 1 to r where r is the rank of V_{PB}. The value of r is different for different configurations as illustrated in Tables II–VI. As can be seen in these tables, the value of r is $c-1$ for configurations ($G \times 1$) where c is the number of classes in the original data. This is the case where the proposed F-LDA simplifies to the conventional LDA. Since the proposed approach allows the use of different configurations $G \times B$ which leads to different classification accuracies as will be shown in Section V, this article therefore empirically exploits different cases (configurations) and highlights the one that gives the best classification result as the optimal parameters.

$$V_W = \sum_{j=1}^{c} \sum_{i=1}^{N_j} \begin{bmatrix} (p_1 - m_{j1})(p_1 - m_{j1}) & (p_1 - m_{j1})(p_2 - m_{j2}) & \cdots & (p_1 - m_{j1})(p_G - m_{jG}) \\ (p_2 - m_{j2})(p_1 - m_{j1}) & (p_2 - m_{j2})(p_2 - m_{j2}) & \cdots & (p_2 - m_{j2})(p_G - m_{jG}) \\ \vdots & \vdots & \ddots & \vdots \\ (p_G - m_{jG})(p_1 - m_{j1}) & (p_G - m_{jG})(p_2 - m_{j2}) & \cdots & (p_G - m_{jG})(p_G - m_{jG}) \end{bmatrix} \tag{17}$$

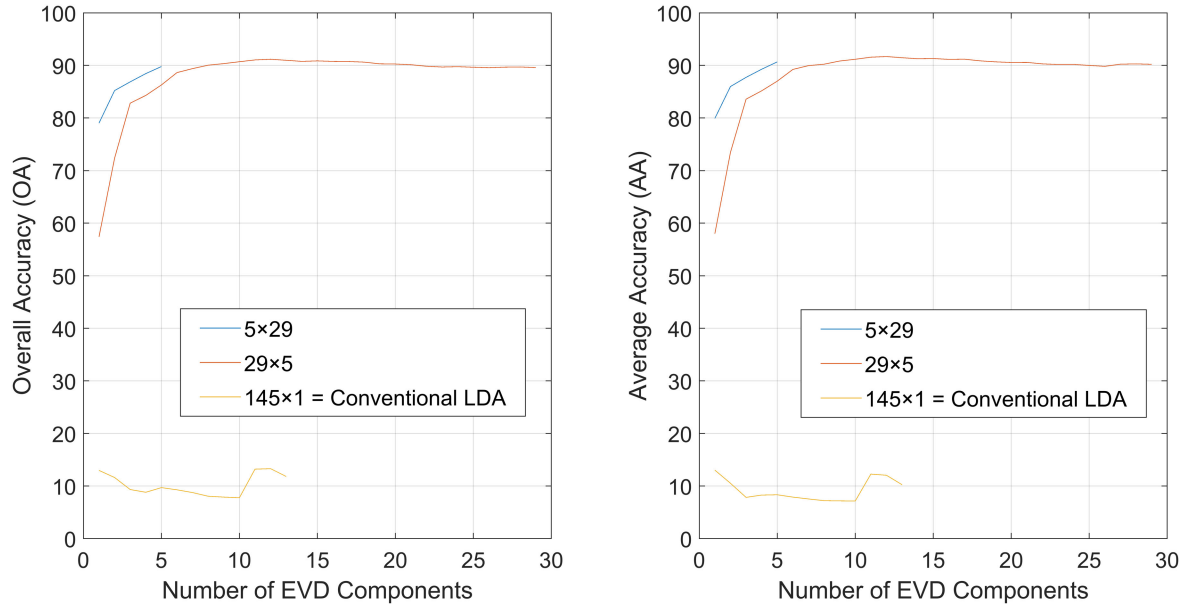


Fig. 3. Classification results for the Botswana dataset using F-LDA.

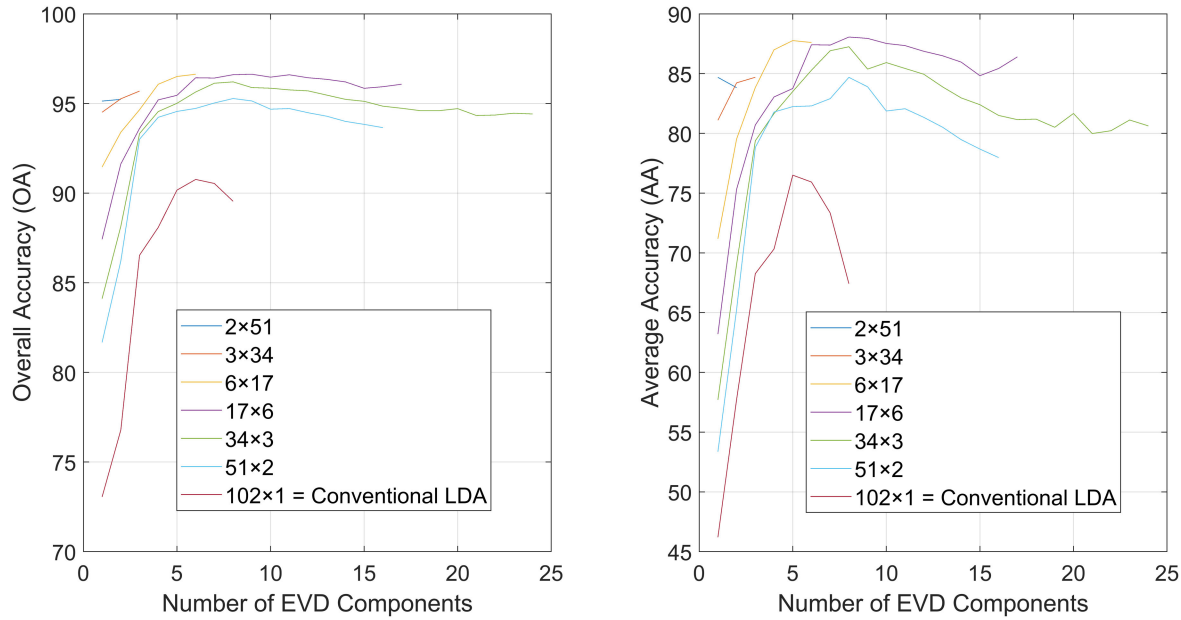


Fig. 4. Classification results for the Pavia Center dataset using F-LDA.

In a situation where f is a prime number, the configuration of the folded matrices is in principle limited to $f \times 1$ and $1 \times f$. In order to explore other configurations ($G \times B$) while applying our proposed F-LDA on the data, additional feature vectors of zeros can be added to the data so that the empty spaces in the folded matrices can be filled with zeros [34].

E. Classification

Support vector machine (SVM) is the machine learning model selected for classification and performance comparison of the proposed and the traditional approaches. The SVM finds an optimal hyperplane in higher dimensional space using some kernel functions to discriminate the different categories in the data. SVM is adopted (and implemented in this article using the Radial bias function) because of the satisfactory classification performance it achieved in related works [54], [55].

The SVM classifier is trained using k-fold cross validation ($k = 5$) to determine the optimal values of the parameters, the penalty (c) and the gamma (g) using a grid search. The optimal value of c and g are then used to obtain the SVM model for final evaluation on the test set. This experiment is repeated ten

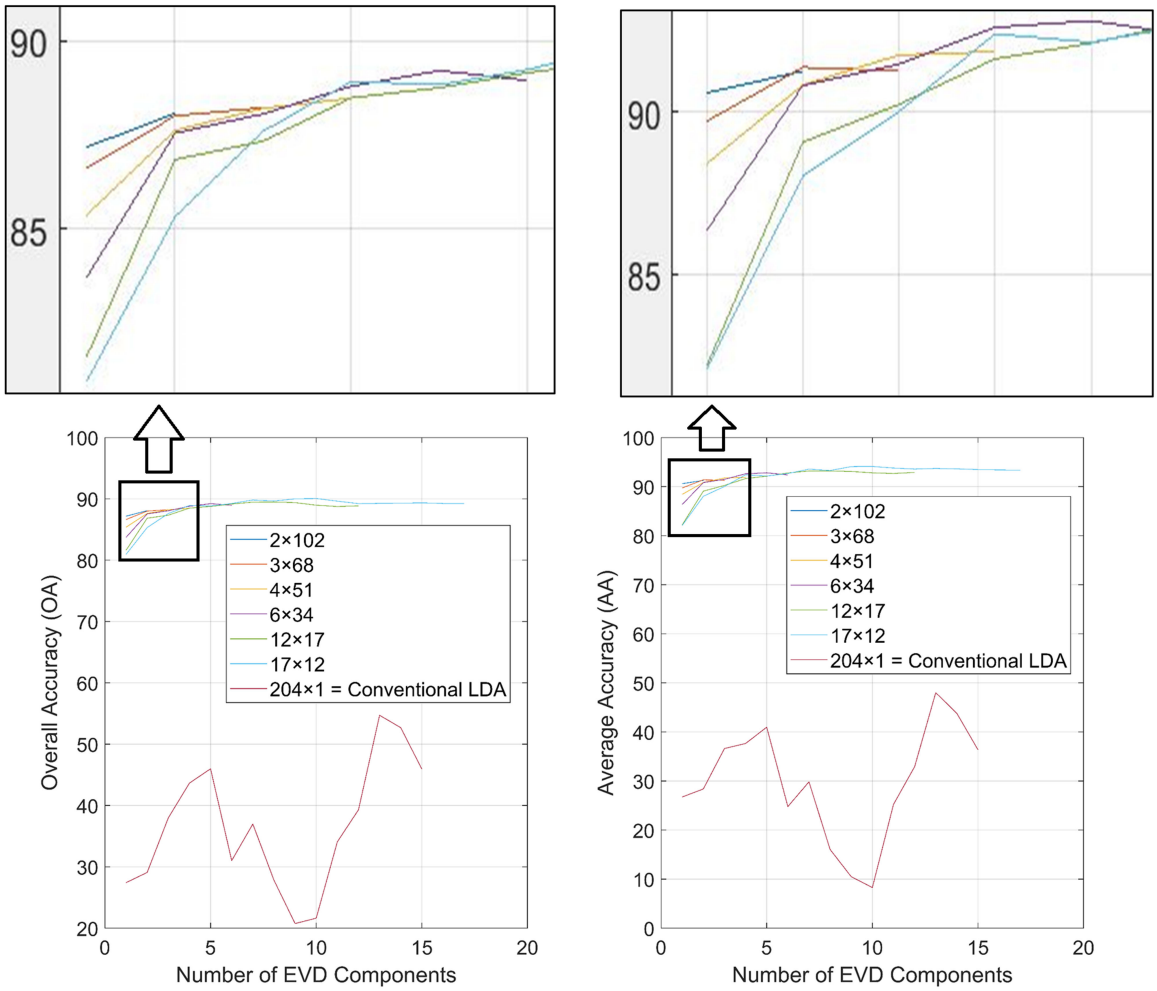


Fig. 5. Classification results for the Salinas dataset using F-LDA.

TABLE VII
COMPUTATIONAL COMPLEXITY FOR THE DIFFERENT STAGES OF THE PROPOSED F-LDA AND THE CONVENTIONAL LDA

Stages	LDA	F-LDA	Saving factors
Within-class variance matrix	$o(cN_jG^2B^2)$	$o(cN_jG^2B)$	B
Between-class variance matrix	$o(cG^2B^2)$	$o(cG^2B)$	B
Transformation matrix	$o(G^3B^3)$	$o(G^3)$	B^3
Eigen problem	$o(G^3B^3)$	$o(G^3)$	B^3
Data projection	$o(sGBd)$	$o(sGd)$	B

times and the classification results recorded in all the cases are averaged and reported.

IV. DATASETS AND EXPERIMENTAL SETTINGS

A. Datasets

Five publicly available [56] and widely used datasets [12], [18], [43], were selected for the performance evaluation of the proposed technique. Each of the five datasets was split into training and testing sets. In each case, the ratio of the training and the testing sets was selected to simulate a SSS scenario for training [43], [44]. The description of the selected datasets is provided in the following sections.

1) Botswana: The Botswana hyperspectral data, with a spatial dimension of 1476 × 256 pixels, were captured at the Okavango Delta over the range of 400–2500 nm of the acquiring Hyperion sensor on the NASA EO-1 satellite. There were 242 spectral bands in the data which contains 14 different classes. A total of 97 uncalibrated and noisy bands were discarded while the remaining 145 spectral bands were retained. In this article, the data was split into training (5%) and testing (95%) sets.

2) Pavia Center: The Pavia Center hyperspectral data, with a spatial dimension of 1096 × 1096 pixels, were captured by ROSIS sensor over in Pavia, northern Italy. There were 114 spectral bands in the data which has a geometric resolution of 1.3 m and contains 9 different classes. The number of noisy bands which were discarded is 12 while the remaining 102 spectral bands were retained. In the experiments, 220 samples are selected for training while the rest are used for testing.

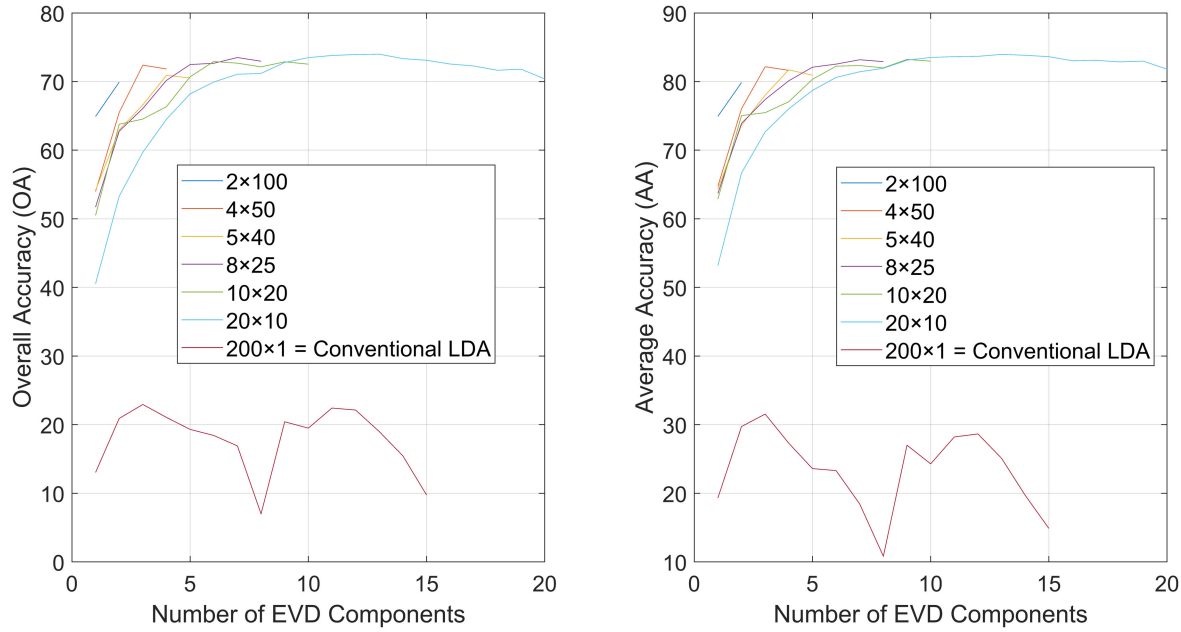


Fig. 6. Classification results for the Indian Pine dataset using F-LDA.

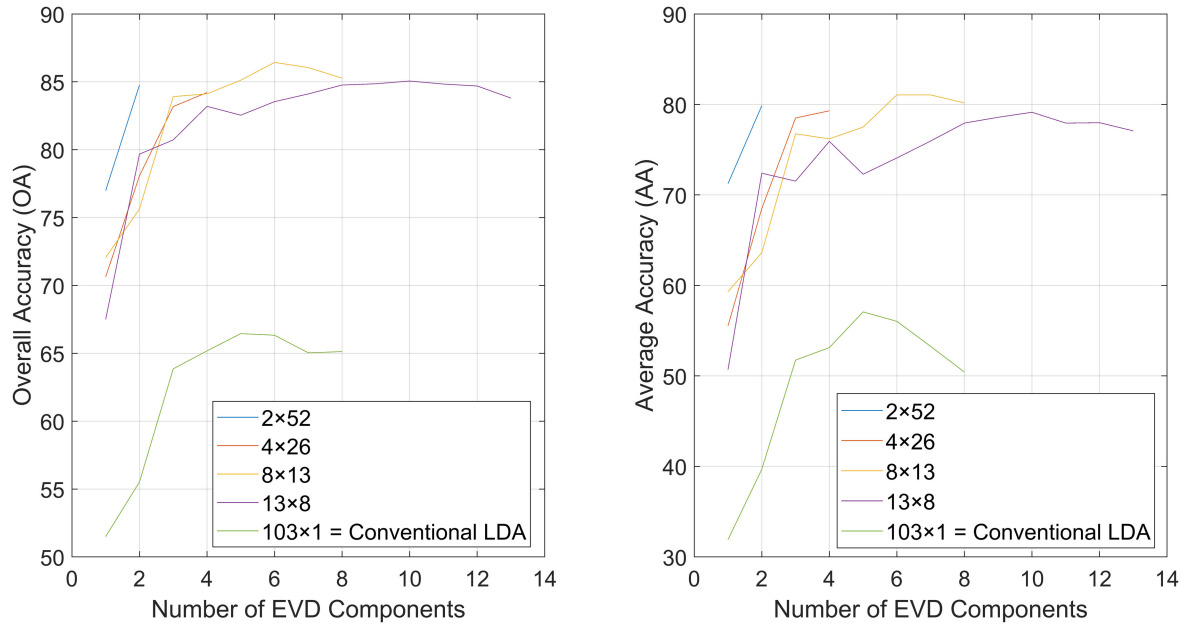


Fig. 7. Classification results for the Pavia University dataset using F-LDA.

3) Salinas: The Salinas hyperspectral image, which was captured by AVIRIS sensor over the Salinas Valley in California, has a spatial dimension of 512 $\times$ 217 pixels and contains 224 spectral bands. A total of 20 noisy bands were discarded, and the remaining 204 spectral bands retained. There are 16 different classes in the data, where 250 samples are selected from the data for training and the rest for testing.

4) Indian Pine: The Indian Pine hyperspectral data, with a spatial dimension of 145 $\times$ 145 pixels, were captured at the Indian Pine test site in North-western Indiana over the range of 400-2500 nm of the acquiring AVIRIS sensor. There were 224 spectral bands in the data containing 16 different classes. A total of 24 noisy bands were discarded while the remaining 200 spectral bands were retained. For the Indian Pine data, 16 samples were selected from each of the 16 classes (totaling 256 samples) for training while the rest are used for testing.

5) Pavia University: The Pavia University hyperspectral image was captured by ROSIS sensor over Pavia, northern Italy. The data has a spatial dimension of 610 $\times$ 340 pixels and a geometric resolution of 1.3 m. There are nine different classes

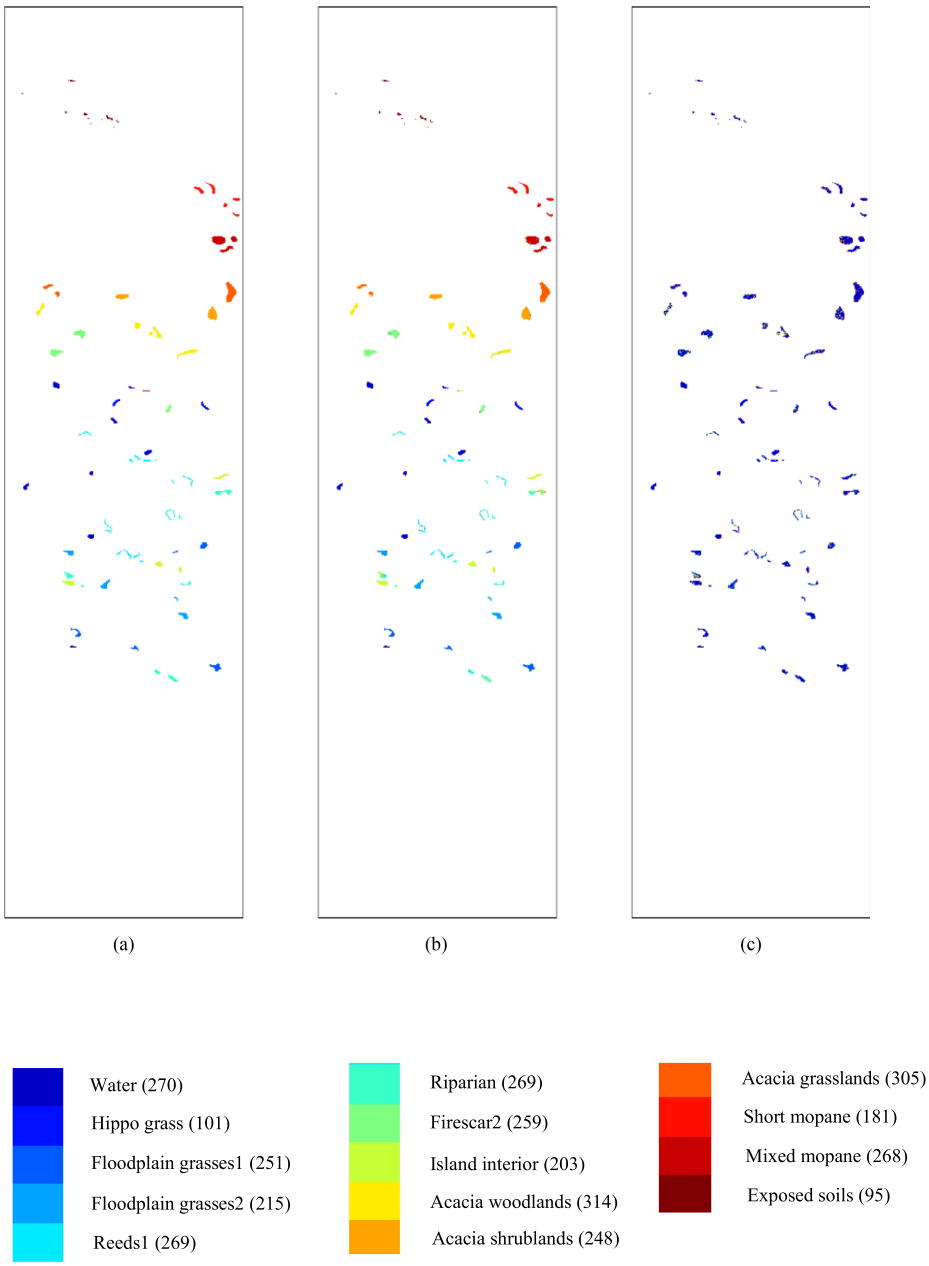


Fig. 8. Botswana data's. (a) Ground truth image. (b) Classification map using F-LDA (29×5). (c) Classification map using LDA (also showing the number of samples in each class).

in the data. From the 115 spectra bands which were present in the data, 12 noisy bands were discarded and the remaining 103 bands retained. A total of 220 samples are selected from the data for training and the rest are selected for testing.

B. Experimental Settings

In this article, we adopted overall accuracy (OA), average accuracy (AA) and kappa coefficient (k), which are widely used in related fields [23], [32], [44] as metrics for performance evaluation of our proposed technique on the five publicly available datasets previously described. To train the SVM classifier on each of the datasets, we initially used three different feature schemes: original feature space; LDA features; and F-LDA features. The number of LDA features d extracted from each dataset is varied from 1 up to the number of class in the dataset minus 1 ($c-1$). For our proposed F-LDA, as explained in Section III-D, the number of features d extracted is dependent on the values of B and d_{EVD} and is given as $B \times d_{\mathrm{EVD}}$ where d_{EVD} is the number of EVD components selected for projection (prior to unfolding of the projected data).

For F-LDA, the number of features is also varied from 1 up to r, the rank of the between-class variance matrix, V_{PB}. For the particular case of the Pavia University dataset, the number of features is 103 which is a prime number and the size of the converted matrices that can be obtained is limited to 1×103 and

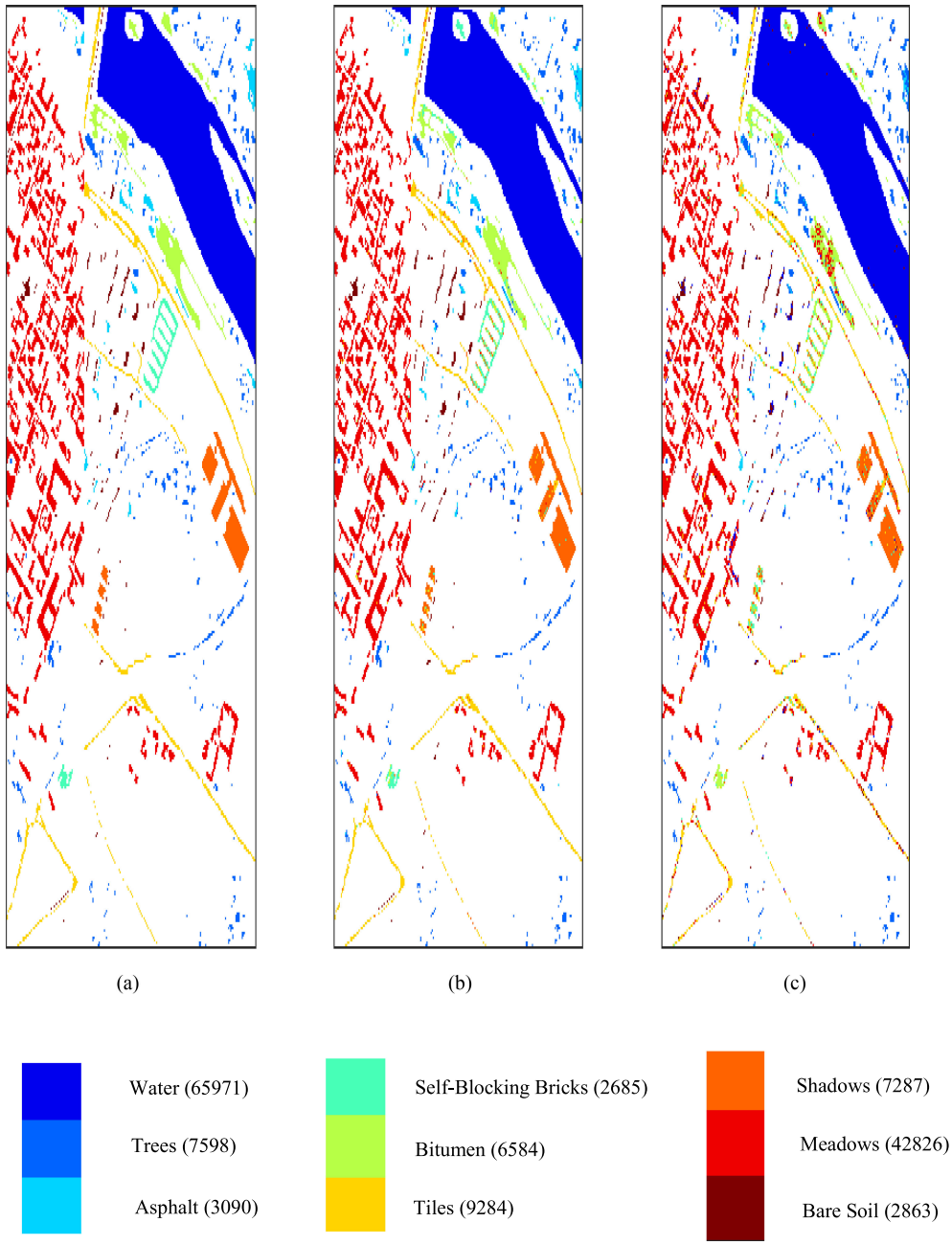


Fig. 9. Pavia Center data's. (a) Ground truth image. (b) Classification map using F-LDA (17×6). (c) Classification map using LDA (also showing the number of samples in each class).

103×1, therefore, to apply our proposed F-LDA on the Pavia University dataset using other configurations (see Table VI), additional zeros are added to the data so that the empty spaces in the converted matrices can be filled [33], [34].

The performance of the proposed approach is compared with that of the 2-D LDA [49], GDA [21], and NWFE [22]. This is necessary for fair comparisons since 2-D LDA, GDA and NWFE are supervised techniques like the proposed approach. The proposed approach is also compared with two unsupervised techniques, namely KPCA [32] and F-PCA [34]. For GDA and KPCA, the Gaussian kernel is selected as the kernel function and its parameter (width) optimized in the range $[10^1,\ 10^2, \ldots,\ 10^5]$ [21], [57]. Different configurations of the converted matrices are also exploited when the 2-D LDA and F-PCA are applied on the five datasets for comparison with the proposed approach. In the 2-D LDA, data projection was done using $y = {P_n}^T v_{Pd}, P_n \in \Re^{G \times B}, v_{Pd} \in \Re^{G \times 1}$, where v_{Pd} is the single projection vector and so the number of features that can be extracted is limited to B, the number of columns in the converted matrices [49].

V. Experimental Results and Analysis

In this section, we investigate the effect of our proposed method on the classification accuracy, computational complexity, and contiguous memory requirement for all five datasets

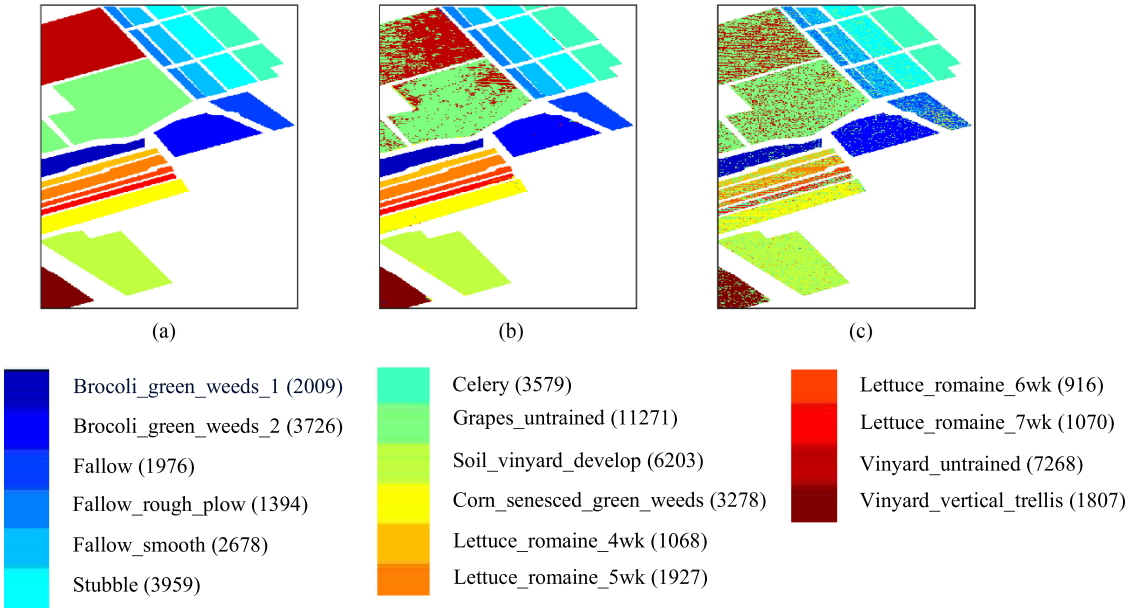


Fig. 10. Salinas data's. (a) Ground truth image. (b) Classification map using F-LDA (17×12). (c) Classification map using LDA (also showing the number of samples in each class).

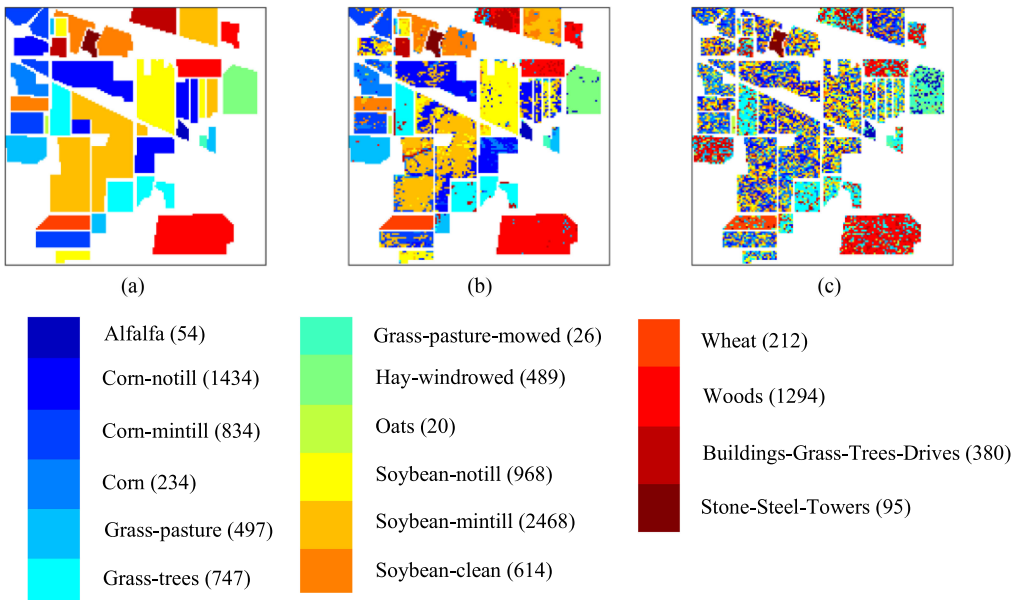


Fig. 11. Indian Pine data's. (a) Ground truth image. (b) Classification map using F-LDA (20×10). (c) Classification map using LDA (also showing the number of samples in each class).

described in the previous section. The experimental results and analysis are therefore presented in three separate sections addressing each of these aspects.

A. *Effect on Classification Accuracy*

First, we trained the SVM classifier using the original feature space available in each of the five datasets. Second, we applied traditional LDA to reduce the dimensionality of the five datasets and used the extracted LDA features to train the SVM classifier. For each dataset, we obtained plots of the OA and AA against the number of features extracted from the LDA and illustrate the results in some figures. We then selected the maximum OA and maximum AA from these plots and present these alongside the results of using the original feature space in Tables II to VI. The maximum kappa coefficients (k) in each of the considered cases are given in Tables II–VI. As can be seen in these tables, the classification accuracy of the SVM classifier is lower when it was trained with the LDA features than when the original feature space was utilized to train the SVM classifier for all the five

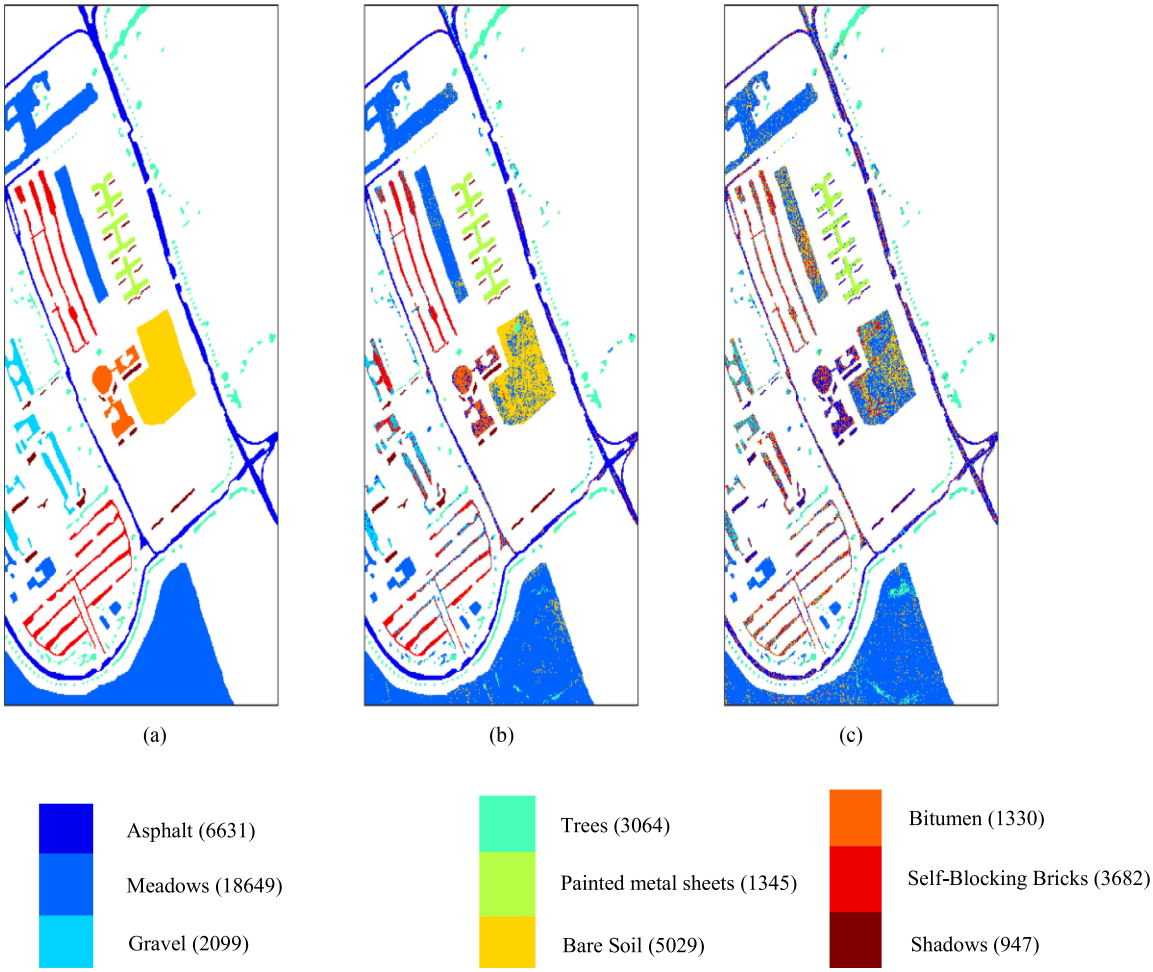


Fig. 12. Pavia University data's. (a) Ground truth image. (b) Classification map using F-LDA (8×13). (c) Classification map using LDA (also showing the number of samples in each class).

datasets considered. These results are not unexpected since LDA is known for producing suboptimal performance when applied in SSS scenarios [18], [20], [43]–[45]. Finally, we applied our proposed F-LDA to reduce the dimensionality of the five datasets and used the outputs to train the SVM classifier, comparing with the conventional LDA, and also other methods including 2-D LDA, GDA, NWFE, KPCA, and F-PCA.

1) Classification Accuracy for the Botswana Dataset: We applied F-LDA on the Botswana dataset and exploited different configurations ($G \times B$). For each of the configurations, we obtained plots of the OA and AA against the number of components at the EVD, d_{EVD} and illustrate these in Fig. 3. We then extracted the maximum OA and AA from each of these plots and present the classification results in Table II. From Table II, we observed that OA, AA, and k were lowest when we set the configuration to 145×1. This is the F-LDA case where $G = f$, which simplifies to the conventional LDA. Also, from Table II, we observed that the maximum OA, AA and k was highest when we set the F-LDA configuration to 29×5. This is an improvement on the classification accuracy with relation to not only the conventional LDA, but also to the case when the SVM classifier was trained using the original feature space. At the same time, the case where $G = 1$ simplifies our proposed F-LDA to the original feature space. We went further to train the SVM classifier using the outputs of the 2-D LDA, GDA, NWFE, KPCA, and F-PCA on the Botswana dataset and present the best OA, AA and k given in Table II. From this table, one can see that the proposed approach consistently gives better OA, AA and k than the other techniques used to benchmark its performance.

2) Classification Accuracy for the Pavia Center Dataset: We repeated the F-LDA operation on the Pavia Center dataset. The plots of OA and AA against d_{EVD} obtained for each of the configurations exploited when the Pavia center dataset was used are presented in Fig. 4. We extracted the maximum OA and AA from Fig. 4 and present the classification results including the maximum k in Table III. We observed that the highest OA, AA, and k were obtained when we set the configuration to 17×6, improving the classification accuracy obtained when the SVM classifier was trained using the original feature space. The case 102×1 simplifies the proposed F-LDA to the conventional LDA, while when $G = 1$ (i.e., 1×102), it simplifies to the original feature space. We also used the features extracted when we applied the 2-D LDA, GDA, NWFE, KPCA and F-PCA on the Pavia Center dataset to train the SVM classifier and

TABLE VIII
COMPUTATIONAL COMPLEXITY (CONTENT CONSUMPTION) FOR THE DIFFERENT STAGES OF THE PROPOSED F-LDA AND THE CONVENTIONAL LDA

Datasets		Best configuration ($G \times B$)	Within-class variance matrix	Between-class variance matrix	Transformation matrix	Eigen problem	Data projection
Botswana	LDA	N/A	o($294350N_j$)	o(294350)	o(3048625)	o(3048625)	o($145\ sd$)
	F-LDA	29 × 5	o($58870N_j$)	o(58870)	o(24389)	o(24389)	o($29\ sd$)
	Saving factors	-	5	5	125	125	5
Pavia Center	LDA	N/A	o($93636N_j$)	o(93636)	o(1061208)	o(1061208)	o($102\ sd$)
	F-LDA	17×6	o($15606N_j$)	o(15606)	o(4913)	o(4913)	o($17\ sd$)
	Saving factors	-	6	6	216	216	6
Salinas	LDA	N/A	o($665856N_j$)	o(665856)	o(8489664)	o(8489664)	o($204sd$)
	F-LDA	17 × 12	o($55488N_j$)	o(55488)	o(4913)	o(4913)	o($17sd$)
	Saving factors	-	12	12	1728	1728	12
Indian Pine	LDA	N/A	o($640000N_j$)	o(640000)	o(8000000)	o(8000000)	o($200sd$)
	F-LDA	20×10	o($64000N_j$)	o(64000)	o(8000)	o(8000)	o($20sd$)
	Saving factors	-	10	10	1000	1000	10
Pavia University	LDA	N/A	o($97344N_j$)	o(97344)	o(1124864)	o(1124864)	o($104\ sd$)
	F-LDA	8 × 13	o($7488N_j$)	o(7488)	o(512)	o(512)	o($8\ sd$)
	Saving factors	-	13	13	2197	2197	13

TABLE IX
FEATURE EXTRACTION TIME (SECONDS) OF DIFFERENT TECHNIQUES (USING THE FIRST FIVE EVD COMPONENTS WHEN APPLICABLE, F-LDA AND 2-D LDA INCLUDE RELATED CONFIGURATION)

Techniques	Botswana	Pavia Center	Salinas	Indian Pine	Pavia University
LDA	0.012	0.072	0.063	0.027	0.029
F-LDA	0.084 (29×5)	2.059 (17×6)	1.024 (17×12)	0.221 (20×10)	0.432 (8×13)
2D LDA	0.032 (1×145)	0.929 (1×102)	0.595 (1×204)	0.119 (1×200)	0.292 (1×103)
NWFE	0.111	0.085	0.233	0.248	0.076
GDA	4.846	267.021	91.95	17.732	76.052
F-PCA	0.179	1.654	0.991	0.244	1.028
KPCA	0.021	0.860	0.423	0.096	0.250

TABLE X
DIFFERENT STAGES OF THE F-LDA AND LDA AND CORRESPONDING MEMORY REQUIREMENTS

Stages	LDA	F-LDA	Saving factor
Data matrix size	$s \times GB$	$G \times B$	s
Within-class variance matrix size	$GB \times GB$	$G \times G$	B^2
Between-class variance matrix size	$GB \times GB$	$G \times G$	B^2
Transformation matrix size	$GB \times GB$	$G \times G$	B^2
Projection matrix size	$GB \times d$	$G \times {}^{d}/_{B}$	B^2

present the classification results given in Table III. It can be seen that the proposed approach continues to give the best OA, AA, and *k*.

3) Classification Accuracy for the Salinas Dataset: We also applied the proposed F-LDA on the Salinas data and present the plots of OA and AA against d_{EVD} obtained for each of the configurations exploited in Fig. 5. We went on to extract the maximum OA and AA from these plots and present the classification results including the maximum *k* in Table IV. For the Salinas data, as can be seen in Table IV, the highest OA, AA and *k* were achieved when the configuration was set to 17×12 (90.06%), higher than the 87.20% achieved when using the original feature space to train the SVM classifier. Again, our proposed F-LDA simplifies to the conventional LDA when the configuration is set to 204×1 (i.e., when $G = f$), and to the original feature space when $G = 1$. Our F-LDA gives the best OA, AA, and *k* as can given in Table IV.

4) Classification Accuracy for the Indian Pine Dataset: Fig. 6 presents the plots of OA and AA against d_{EVD} for each of the configurations exploited when the F-LDA operation was performed on the Indian Pine dataset, with related classification results in Table V. Lowest OA, AA and *k* were obtained for the

configuration 200×1. The highest OA, AA and *k* were attained when we set the configurations to 20×10. The highest OA, AA and *k* reported for the F-LDA are much higher than those attained when we trained the SVM classifier using the features extracted from the conventional LDA. As in the other three datasets, the extreme case 200×1 simplifies the proposed F-LDA to conventional LDA while the other extreme case 1×200 simplifies the proposed F-LDA to the original feature space. We also applied the 2-D LDA, GDA, NWFE, KPCA and F-PCA to reduce the dimensionality of the Indian Pine dataset, where our proposed approach achieves the best performance again.

5) Classification Accuracy for the Pavia University Dataset: Finally, we also apply the outputs of the F-LDA on the Pavia University dataset to train the SVM classifier and used the classification results attained to obtain the plots of OA and AA against d_{EVD} which are presented in Fig. 7, with highest classification values reported and compared in Table VI. The highest OA, AA and *k* were obtained for the configuration 8×13 , improving accuracies from the original feature space. For this dataset, configurations 103×1 and 1×103 simplify the proposed F-LDA to conventional LDA and to the original feature space, respectively. We also used the features extracted when we applied the 2-D LDA, GDA, NWFE, KPCA and F-PCA on the Pavia University data to train the SVM classifier and present the classification results obtained in Table VI. It can be seen in Table VI that the proposed approach continues to give the best OA, AA, and *k*.

Finally, we present the classification maps of all hyperspectral data in Figs 8–12 to allow visualization and qualitative analysis of the proposed F-LDA approach compared to traditional LDA features when used for classification. As can be seen in Figs 8–12, the classification maps obtained using the F-LDA are, in general, smoother than those obtained using the LDA for all the five datasets used.

B. Effect on Computational Complexity

We illustrate and compare the computational complexity of the different stages of the conventional LDA and our proposed F-LDA in Table VII where c, N_j and d are the number of classes in the data, the number of samples in each class and the number of features extracted respectively.

In the conventional LDA, the computational complexity of calculating the within-class variance V_W and between-class variance V_B matrices in (3) and (4) are $o(cN_jG^2B^2)$ and $o(cG^2B^2)$, respectively, where $V_W \in \Re^{GB\times GB}$, $V_B \in \Re^{GB\times GB}$, $(x_{ij} - m_j) \in \Re^{1\times GB}$ and $(m_j - m) \in \Re^{1\times GB}$. The complexity of computing ${V_W}^{-1}$ is $O(G^3B^3)$. $O(G^3B^3)$ is also the computational complexity of multiplying ${V_W}^{-1}$ and V_B. Hence, to compute the transformation matrix T and eigenvectors, the required computational complexity is $o(G^3B^3)$. The computational complexity of projecting the data using (6) is $o(sGBd)$.

In the proposed F-LDA, the computational complexity of calculating the within-class variance V_{PW} and between-class variance V_{PB} matrices in (11) and (12) are $o(cN_jBG^2)$ and $o(cBG^2)$, respectively, where $V_{PW} \in \Re^{G\times G}$, $V_{PB} \in \Re^{G\times G}$, $(P_{ij} - M_j) \in \Re^{G\times B}$ and $(M_j - M) \in \Re^{G\times B}$. The computational complexity of calculating ${V_{PW}}^{-1}$ is *o(G^3).* To compute the product of ${V_{PW}}^{-1}$ and V_{PB}, the required computational complexity is *o(G^3).* Hence, to compute the transformation matrix T and eigenvectors, the computational complexity is $o(G^3)$. To project each sample using (14), the required complexity is Gd where $d = B \times d_{EVD}$. Hence, the computational complexity of projecting all the samples is sGd.

We also used the computational complexities presented in Table VII to compute the content consumption for each dataset and present the results in Table VIII. As can be seen in Tables VII and VIII, F-LDA requires less computational complexity to implement all stages. For the within-class variance and between-class variance computation, the complexity is reduced by a saving factor of B. A saving factor of B^3 is reported for the transformation matrix and eigenvectors computation. A reduction in the cost of projecting the data by B is reported. This reduction in the computational complexity is achieved thanks to the new dimensions of the within-class variance, between-class variance, and the transformation matrices in the proposed approach which are now smaller than their counterparts in the traditional approach. Instead of processing big matrices, we now have a set of smaller ones to deal with.

We also present the feature extraction time of the proposed approach on the datasets (using the first five EVD components when applicable) and compare to the other techniques in Table IX. In all cases, it can be seen that the conventional LDA is faster than both the 2-D LDA and F-LDA. This is due to the additional time used by the 2-D LDA and F-LDA for the feature vector—feature matrix conversion. F-LDA can also be seen to be slightly slower than the 2-D LDA in all cases because of the additional time to consider the eigenvectors individually and unfold the projected samples. The F-LDA can also be seen to be slower than the K-PCA but faster than the NWFE, GDA, and F-PCA in some cases. Overall, the range of the feature extraction time reported for the proposed F-LDA is 0.084–2.059 s, slightly slower than its counterparts but negligible considering its higher classification accuracy.

C. Effect on Contiguous Memory Requirement

We illustrate the contiguous memory requirement for the different stages of the conventional LDA and our proposed F-LDA in Table X . We went further to use the results presented in Table X to compute the content consumption for each dataset and illustrate these in Table XI. From Tables X and XI, we observe that the memory required for the within-class variance matrix, between-class variance matrix and transformation matrix is less in F-LDA than in LDA by a saving factor of B^2. Also, for the original data matrix and the projected data matrix, saving factors of S and B^2 are achieved respectively when the F-LDA is utilized instead of the LDA. In general, the contiguous memory required for the F-LDA is much less than what is required for the conventional LDA. This is mainly due to the dimension of the matrices at the different stages of the F-LDA which are now smaller than those in the conventional LDA.

TABLE XI
DIFFERENT STAGES OF THE F-LDA AND LDA AND CORRESPONDING MEMORY REQUIREMENTS (CONTENT CONSUMPTION)

Datasets		Best Configuration ($G \times B$)	Data matrix size	Within-class variance matrix size	Between-class variance matrix size	Transformation matrix size	Projection matrix size
Botswana	LDA	-	145s	21025	21025	21025	145d
	F-LDA	29 × 5	145	841	841	841	5.8d
	Saving factors	-	s	25	25	25	25
Pavia Center	LDA	-	102s	10404	10404	10404	102d
	F-LDA	17×6	102	289	289	289	2.8333d
	Saving factors	-	s	36	36	36	36
Salinas	LDA	-	204s	41616	41616	41616	204d
	F-LDA	17 × 12	204	289	289	289	1.4167d
	Saving factors	-	s	144	144	144	144
Indian Pine	LDA	-	200s	40000	40000	40000	200d
	F-LDA	20×10	200	400	400	400	2d
	Saving factors	-	s	100	100	100	100
Pavia Univ.	LDA	-	104s	10816	10816	10816	104d
	F-LDA	8 × 13	104	64	64	64	0.6154d
	Saving factors	-	s	169	169	169	169

VI. CONCLUSION

This article presents an F-LDA for effective and efficient feature extraction and dimensionality reduction of remotely sensed hyperspectral data in SSS scenarios. The proposed F-LDA replicates a simple, but effective mathematical "trick" (folding the pixels) which was motivated by previous work to extend PCA [34]. Now more informative features are produced by the F-LDA (higher classification accuracy than the original feature space, conventional LDA, 2-D LDA, and other state-of-the-art methods) with reduced contiguous memory requirement and reduced complexity. The performance of the proposed technique is evaluated on five publicly available datasets from different sensors (AVIRIS, ROSIS, Hyperion) and the experimental results demonstrate the superiority of the proposed technique to the traditional approach when applied in SSS scenarios. Future work will focus on proposing novel techniques to automatically select the best parameters for the F-LDA configuration, including combinations of G and B which lead to sparse 2-D matrices, and exploring other related improvements, such as incorporating spatial information and recent advances in deep learning.

REFERENCES

[1] H. Fu, G. Sun, J. Zabalza, A. Zhang, J. Ren, and X. Jia, "A novel spectral-spatial singular spectrum analysis technique for near real-time in situ feature extraction in hyperspectral imaging," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2214–2225, May 2020, doi: 10.1109/JSTARS.2020.2992230.

[2] 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, Jan./Dec. 2017.

[3] Q. Yang, Y. Xu, Z. Wu, and Z. Wei, "Hyperspectral and multispectral image fusion based on deep attention network," in *Proc. Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, 2019, pp. 1–5.

[4] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4117–4128, Sep. 2016.

[5] K. Karalas, G. Tsagkatakis, M. Zervakis, and P. Tsakalides, "Land classification using remotely sensed data: Going multilabel," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3548–3563, Jun. 2016.

[6] A. Ferreira *et al.*, "Eyes in the skies: A data-driven fusion approach to identifying drug crops from remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4773–4786, Dec. 2019.

[7] Y. Zhang, G. Cao, A. Shafique, and P. Fu, "Label propagation ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3623–3636, Sep. 2019.

[8] X. Wei, W. Zhu, B. Liao, and L. Cai, "Matrix-based margin-maximization band selection with data-driven diversity for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7294–7309, Dec. 2018.

[9] S. D. Fabiyi *et al.*, "Comparative study of PCA and LDA for rice seeds quality inspection," in *Proc. IEEE AFRICON*, 2019, pp. 1–4.

[10] A. Zhang *et al.*, "Hyperspectral band selection using crossover-based gravitational search algorithm," *IET Image Process*, vol. 13, no. 2, pp. 280–286, Feb. 2019.

[11] J. Zabalza *et al.*, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4418–4433, Aug. 2015.

[12] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[13] X. Dai and W. Xue, "Hyperspectral remote sensing image classification based on convolutional neural network," in *Proc. Chin. Control Conf.*, 2018, vol. 2018, pp. 10373–10377.

[14] J. Leng, T. Li, G. Bai, Q. Dong, and H. Dong, "Cube-CNN-SVM: A novel hyperspectral image classification method," in *Proc. IEEE 28th Int. Conf. Tools Artif. Intell.*, 2016, pp. 1027–1034.

[15] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

[16] M. Kamandar and H. Ghassemian, "Linear feature extraction for hyperspectral images based on information theoretic learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 702–706, Jul. 2013.

[17] H. Yu, L. Gao, J. Li, S. Li, B. Zhang, and J. Benediktsson, "Spectral-spatial hyperspectral image classification using subspace-based support vector machines and adaptive markov random fields," *Remote Sens.*, vol. 8, no. 4, pp. 1–21, Apr. 2016.

[18] W. Liao, A. Pižurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.

[19] X. Yin, R. Wang, X. Liu, and Y. Cai, "Deep forest-based classification of hyperspectral images," in *Proc. Chin. Control Conf.*, 2018, vol. 2018, pp. 10367–10372.

[20] H. Yuan, Y. Lu, L. Yang, H. Luo, and Y. Y. Tang, "Spectral-spatial linear discriminant analysis for hyperspectral image classification," in *Proc. IEEE Int. Conf. Cybern.*, 2013, pp. 144–149.

[21] G. Yang, X. Yu, and X. Zhou, "Hyperspectral image feature extraction based on generalized discriminant analysis," in *Proc. 21st ISPRS Congr.*, 2008, pp. 285–290.

[22] B. C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[23] B. Tu, J. Wang, G. Zhang, X. Zhang, and W. He, "Texture pattern separation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3602–3614, Sep. 2019.

[24] J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1886–1890, Nov. 2014.

[25] J. Wang and C. I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.

[26] R. Reshma, V. Sowmya, and K. P. Soman, "Effect of Legendre–Fenchel denoising and SVD-based dimensionality reduction algorithm on hyperspectral image classification," *Neural Comput. Appl.*, vol. 29, no. 8, pp. 301–310, Apr. 2018.

[27] M. Huang, Q. Zhu, B. Wang, and R. Lu, "Analysis of hyperspectral scattering images using locally linear embedding algorithm for apple mealiness classification," *Comput. Electron. Agriculture*, vol. 89, pp. 175–181, Nov. 2012.

[28] W. Li, L. Zhang, L. Zhang, and B. Du, "GPU parallel implementation of isometric mapping for hyperspectral classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1532–1536, Sep. 2017.

[29] V. Menon, Q. Du, and J. E. Fowler, "Fast SVD with random hadamard projection for hyperspectral dimensionality reduction," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 9, pp. 1275–1279, Sep. 2016.

[30] J. Xia, P. Du, X. He, and J. Chanussot, "Hyperspectral remote sensing image classification based on rotation forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 239–243, Jan. 2014.

[31] 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.

[32] M. Fauvel, J. Chanussot, and A. Benediktsson, "Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas," *EURASIP J. Adv. Signal Process.*, vol. 783194, pp. 1–14 2009.

[33] J. Zabalza, J. Ren, Z. Wang, H. Zhao, J. Wang, and S. Marshall, "Fast implementation of singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2845–2853, Jun. 2015.

[34] J. Zabalza *et al.*, "Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 112–122, 2014.

[35] R. Hang *et al.*, "Robust matrix discriminative analysis for feature extraction from hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 2002–2011, May 2017.

[36] H. Yuan, Y. Y. Tang, Y. Lu, L. Yang, and H. Luo, "Spectral-spatial classification of hyperspectral image based on discriminant analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2035–2043, Jun. 2014.

[37] S. W. Kim and R. P. W. Duin, "On using a pre-clustering technique to optimize LDA-based classifiers for appearance-based face recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4756. New York, NY, USA: Springer, 2007, pp. 466–476.

[38] X. Song, S. Jiang, S. Wang, J. Tang, and Q. Huang, "Cross concept local fisher discriminant analysis for image classification," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7733. New York, NY, USA: Springer, 2013, pp. 407–416.

[39] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, pp. 169–190, 2017.

[40] N. Elhadji, I. Gado, E. Grall-Maës, and M. Kharouf, "Linear discriminant analysis based on fast approximate SVD," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 359–365.

[41] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large scale discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 1–12, Jan. 2008.

[42] C. E. Thomaz, E. C. Kitani, and D. F. Gillies, "A maximum uncertainty LDA-based approach for limited sample size problems - with application to face recognition," *J. Braz. Comput. Soc.*, vol. 12, no. 2. pp. 7–18, 2006.

[43] L. He, H. Yang, and L. Zhao, "Tensor subspace learning and classification: Tensor local discriminant embedding for hyperspectral image," in *Proc. Int. Conf. Comput. Vis. Workshop,* 2019, pp. 589–598.

[44] H. R. Shahdoosti and F. Mirzapour, "Spectral-spatial feature extraction using orthogonal linear discriminant analysis for classification of hyperspectral data," *Eur. J. Remote Sens.*, vol. 50, pp. 111–124, 2017.

[45] M. Imani and H. Ghassemian, "Feature reduction of hyperspectral images: Discriminant analysis and the first principal component," *J. Artif. Intell. Data Mining*, vol. 3, no. 1, pp. 1–9, Jan. 2015.

[46] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1569–1576.

[47] H. Kong, E. K. Teoh, J. G. Wang, and R. Venkateswarlu, "Two dimensional fisher discriminant analysis: Forget about small sample size problem," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2005.

[48] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 527–532, Apr. 2005.

[49] M. Imani and H. Ghassemian, "Two dimensional linear discriminant analyses for hyperspectral data," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 10, pp. 777–786, Oct. 2015.

[50] D. B. Gillis and J. H. Bowles, "An introduction to hyperspectral image data modeling," in *Applied and Numerical Harmonic Analysis*. New York, NY, USA: Springer, 2013, pp. 173–194.

[51] S. D. Fabiyi *et al.*, "Varietal classification of rice seeds using RGB and hyperspectral images," *IEEE Access*, vol. 8, pp. 22493–22505, 2020.

[52] D. W. Sun, *Hyperspectral Imaging for Food Quality Analysis and Control*. Amsterdam, The Netherlands: Elsevier, 2010.

[53] Z. Qiu, J. Chen, Y. Zhao, S. Zhu, Y. He, and C. Zhang, "Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network," *Appl. Sci.*, vol. 8, no. 2, pp. 1–12, Jan. 2018.

[54] B. Tu, C. Zhou, D. He, S. Huang, and A. Plaza, "Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4116–4131, Jun. 2020.

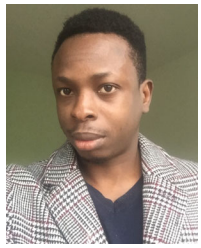
[55] C. Mu, J. Liu, Y. Liu, and Y. Liu, "Hyperspectral image classification based on active learning and spectral-spatial feature fusion using spatial coordinates," *IEEE Access*, vol. 8, pp. 6768–6781, 2020.

[56] M. Graña, M. A. Veganzons, and B. Ayerdi, "Hyperspectral remote sensing," *Comput. Intell. Group*, Accessed: Nov. 27, 2020. [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_scene

[57] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1139–1155, Apr. 2009.

[58] S. A. Hosseini and H. Ghassemian, "Rational function approximation for feature reduction in hyperspectral data," *Remote Sens. Lett.*, vol. 7, no. 2, pp. 101–110, Feb. 2016.

[59] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7905–7916, Jul. 2015.



Samson Damilola Fabiyi (Student Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Ilorin, Ilorin, Nigeria, in 2015, and the M.Sc. degree in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K. in 2018. He is currently working toward the Ph.D. degree with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow , U.K.

Since September 2021, he has been a Teaching Fellow with the School of Computing, University of Leeds, Leeds, U.K. His research interests include image processing, hyperspectral imaging, machine learning and internet of things.

Paul Murray received the M.Eng. and Ph.D. degrees in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 2008 and 2012, respectively.

He is currently a Senior Lecturer with the University of Strathclyde. His research interests include image processing, hyperspectral imaging and analysis, feature extraction, and machine learning

Jaime Zabalza (Member, IEEE) received the M.Eng. degree in industrial engineering from the Universitat Jaume I, Castellón de la Plana, Spain, in 2006, and the M.Sc. degree (with distinction) and the Ph.D. degree in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 2012 and 2015, respectively.

He is currently with the Department of Electronic and Electrical Engineering, University of Strathclyde. His research interests include hyperspectral data analysis as well as signal and image processing in a wide range of applications.

Dr. Zabalza was the recipient of the IET Image and Vision Section prize for best Ph.D. thesis for his work in hyperspectral remote sensing.

Jinchang Ren (Senior Member, IEEE) received the B. E. degree in computer software, the M.Eng. degree in image processing, the D.Eng. degree in computer vision from Northwestern Polytechnical University, Xi'an, China, and the Ph.D. degree in electronic imaging and media Communication from the University of Bradford, Bradford, U.K., in 1992, 1997, 2000, and 2009, respectively.

He is currently a Professor of computing sciences with National Subsea Centre, Robert Gordon University, Aberdeen, U.K. He has authored or coauthored more than 300 peer reviewed journal/conferences articles, and acts as an Associate Editor for several international journals including IEEE TGRS and J. of the Franklin Institute, etc. His research interests focus mainly on hyperspectral imaging, image processing, computer vision, big data analytics, and machine learning.