Pattern Recognition 134 (2023) 109096

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

IDA: Improving distribution analysis for reducing data complexity and dimensionality in hyperspectral images

Dalal AL-Alimi[a,c], Mohammed A.A. Al-qaness[b,c,d,*], Zhihua Cai[a,*], Eman Ahmed Alawamy[e]

[a] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321004, China
[c] Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen
[d] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
[e] School of Mathematics and Statistics, Central South University, Changsha 410083, China

ARTICLE INFO

Article history:
Received 3 August 2022
Revised 16 September 2022
Accepted 4 October 2022
Available online 8 October 20

Keywords:
Feature reduction
Hyperspectral image
Classification
Feature fusion
Feature extraction
Dimensionality reduction

ABSTRACT

Hyperspectral images (HSIs) are known for their high dimensionality and wide spectral bands that increase redundant information and complicate classification. Outliers and mixed data are common problems in HSIs. Thus, preprocessing methods are essential in enhancing and reducing data complexity, redundant information, and the number of bands. This study introduces a novel feature reduction method (FRM) called improving distribution analysis (IDA). IDA works to increase the correlation between related data, decrease the distance between big and small data, and correct each value's location to be inside its group range. In IDA, the input data passes through three stages. Getting rid of outliers and improving data correlation is the first step. The second stage involves increasing the variance. The third is to simplify the data and normalize the distribution. IDA is compared with four popular FRMs in four available HSIs. It is also tested and evaluated in various classification models, including spatial, spectral, and spectral-spatial models. The experimental results demonstrate that IDA performs admirably in enhancing data distribution, reducing complexity, and accelerating performance.

1. Introduction

The development of obtaining remote sensing images and improving the resolution has made a big jump in the field of hyperspectral images (HSI). It has become one of the hottest topics. Although this development brings a lot of advantages and lets the obtained HSIs be used in vast and different fields, the researchers face many difficulties related to the large dimensionality and data computational complexity [1,2]. The large dimensionality of HSIs comes from their broad spectral bands and numerous redundancies. In addition, the computational data complexity happens because of the high spectral resolution, different sample ratios in classes, and high dimensionality of the data. All of these obstacles raise processing complexity and reduce classification precision. Dimensionality reduction is an efficient technique for minimizing the dimensionality, complexity, and processing time of HSI.

Dimensionality reduction methods (DRM) have been one of the essential steps in HSI preprocessing. DRM is categorized into feature selection and feature extraction [3,4]. In the feature selection, the informative features are selected as a subset from the original dataset [5]. Feature extraction uses functional mapping to transform the original features into a new subset containing the most critical information [2,6]. Feature selection aims to maintain the physical meaning of the original data by selecting the effective subset of the existing data. Due to feature extraction works to reduce dimensionality and enhance data learning by feature extraction methods (FEM), it has received much interest. FEM is superior to feature selection at handling noise, complexity, and sparsity in real-world datasets. Commonly, FEM can be used to extract the features in the two most common ways: supervised and unsupervised methods [3,4,7]. Supervised FEM employs labeled information while transferring data [20]; for instance, linear discriminant analysis (LDA) and independent component analysis (ICA) [8]. Because no labeled data is used in unsupervised FEM, it focuses on the data. These methods look for the similarity between pieces of data to determine whether they can be categorized and grouped, for example, multi-dimensional scaling (MDS), singular value decomposition (SVD), principal component analysis (PCA), and isometric mapping (ISOMAP).

Principal component analysis (PCA) is the most commonly used unsupervised feature extraction approach. PCA is a linear data transformation that reduces redundancy while increasing variance.

* Corresponding authors.
E-mail addresses: dalal@cug.edu.cn (D. AL-Alimi), alqaness@zjnu.edu.cn (M.A.A. Al-qaness), zhcai@cug.edu.cn (Z. Cai), emanahmed@csu.edu.cn (E.A. Alawamy).

https://doi.org/10.1016/j.patcog.2022.109096
0031-3203/

The principal components are a collection of linearly uncorrelated observations x of dimension M (they reside in $\mathbb{R}^{M}$) [9,10]. LDA is a supervised linear FEM. Because of the dimensionality curse, LDA can be employed as a feature extraction approach to improve computational efficiency and minimize the amount of overfitting. LDA has a similar general concept to PCA. On the other hand, LDA does not try to find the orthogonal component axes with the most variation in a dataset. Instead, it looks for the feature subspace that best separates classes. In other words, LDA forms two scatter matrices: (1) an in-between-class matrix that calculates the distance between each class's mean and (2) a within-class matrix that computes the distance between each class's mean and the data within that class. Independent component analysis (ICA) is a linear FEM that looks for independent and non-Gaussian features. It is unlike PCA; it focuses on increasing the variance. ICA decomposes the input data into a mixing matrix and the basis coefficient and then chooses the top independent components. ICA strives to standardize data distribution and reduce mutual values to the greatest extent possible [3,11,12].

The singular value decomposition (SVD) [13,14] represents linear algebra and factorizes any matrix into singular vectors and singular values: $\mathcal{X} = u.\sigma.v^{T}$. Where $\mathcal{X}$ is the orginal input data with $(m \times n)$ dimination, u and v are orthogonal matrices with $(m \times m)$ and $(n \times n)$ sizes, and σ is $(m \times n)$ digonal matrix. Iterative numerical algorithms are used to calculate the SVD. However, the resultant matrices may contain complex values, and some matrices may fail to decompose accurately due to the limits of floating-point arithmetic. Thus, SVD works better with more components selection [3]. Imani and Ghassemian introduced feature space discriminant analysis (FSDA) [7]. FSDA first works to maximize the differences between features depending on calculating the covariance of the instance. Then, according to the first step results, the mean of each feature is calculated to get the covariance matrix. The whole feature reduction of this study depends on this covariance matrix, which increases the variance between classes besides reducing the band numbers. In 2018, uniform manifold approximation and projection (UMAP) was introduced by McInnes et al. [15] as a nonlinear dimensional reduction method; based on topological algebra in multi-dimensional space. UMAP is a neighbor graph that captures local structure rather than global structure to find a similar graph in lower dimensions to connect to the neighbor's points, which works well with relative values with low space between them. However, UMAP does not include a scaling process and cannot process noisy data. In conclusion, most dimensionality reduction methods (preprocessing methods) do not work to reduce outliers and the distance between related values. Furthermore, enhancing the data distribution helps obtain a more accurate classification, as shown in the result section.

Many methods in deep learning have been used to extract the features of the HSIs, and they give impressive results in HSIs classification compared to machine learning methods. In [16], after reducing the dimensionality by PCA, inter-spectra difference feature (ISDF) was used to model the relationship between close spectra and the neighbor spectral difference feature (NSDF) to extract statistical information as the relationship inside the spectrum respectively. PCA was then used to create a multi-type spectral spatial feature (MSSF) by fusing the changed features. The SVM was then employed to classify MSSF features. The MSSF approach prioritized neighboring spectrum values while disregarding distantly related variables. The SVM was included in a convolutional neural network (DNN) in [17]. Multiple SVM functions adjusted with the SVM regularization parameter were used to initialize the weights of DNN's hidden layers. This integrated model performed better than the traditional SVM in classifying HSI. Chen et al. devised dual-stream deep architecture (SSDS) to extract single spectral and adjacent spatial features in two separate parts, then fuse them in the third part by fusion-wise pooling for filtering and selecting the discriminative features [18]. SSDS takes a long time and ignores the distantly related values. The multi-view deep autoencoder approach was proposed in [19] to extract only important features, removing noise and revealing a common latent representation. In addition, a semi-supervised graph CNN was employed to incorporate graph topology and local vertex features in the convolutional layers to improve HSI classification by retaining spectral-spatial information. In [1], spatial and spectral kernel generation modules were designed to focus more on extracting salient features of HSIs simultaneously, focusing only on the pixels with a strong correlation. The model of this study [20] consists of two components for extracting spatial and spectral characteristics. The RNN component concentrates on learning inner spectral correlations, whilst the 2D-Convolutional neural network (2D-CNN) component operates in the spatial dimension and focuses on the spatial relationships between neighboring pixels. Many studies used parallel networks of multi-scale CNNs to get more spectral-spatial features, which made computing more complicated and time-consuming [21–23]. For instance, in [24], the proposed classification model contains three multi-scale and dimension branches of CNNs, with a residual block in each branch to improve the feature extraction of the HSI and prevent insufficiency. Although deep learning algorithms deliver excellent performance, some likely increase noise values or generate unrelated ones, while the majority disregard distantly related values. Therefore, these have an undeniable effect on precision and lengthen the training period.

Even though the preprocessing of FEMs and deep learning methods have successfully extracted the features, these methods do not improve well the HSIs' data complexity and distribution. These methods may increase noise and outlier values during the classification. Most preprocessing methods ignore the importance of redistributing the shared data to the correct position and category. The reasons behind all these complexities can be summarized as follows:

- The common HSIs classification problem with multi-class images is the imbalance of samples in all classes, which complicates the extraction and reduction equally for all classes.
- Many FRMs and DLMs ignore the relevance between classes' joint values during the processing time, so some potential discriminant information may be lost, which affects the classification speed and accuracy.
- HSI is known for hundreds of bands, redundant data, wide dimensionality, mixed pixels, and more. All of these complicate both reduction and classification and increase the processing time.

Let $\mathcal{V} = \{v_i : v_i \in \mathcal{R}^{d}\}_{i=1}^{N}$ be the input data with the mean of each instance (m). Enhancing the correlation of $(\mathcal{V})$ generates $(\mathcal{D})$ to extract k subspace features from the $(d \times 1)$ original vector space $(\mathcal{V})$. A transformation matrix $(\mathcal{W})$ is used as a weight to extract the final features: $\mathcal{Y}_{q\times k} = \mathcal{D}_{q\times k} \times (\mathcal{W}_{k\times 1})^{T}$. The proposed method, improving distribution analysis (IDA), is a novel unsupervised feature extraction method. It uses three measures to produce the final extracted features that solve the challenges mentioned above. Using four different HSI datasets, IDA performs better than state-of-the-art FEMs that were compared. The main contributions of IDA are as follows:

- The input data in IDA passes through three stages: the first stage reduces the distance between related data, which helps decrease outlier and mixed data. The second stage is to enhance the variations, which enhance the classification process. The final stage is for distribution improvement and getting the new subspace of features.
- IDA provides the best way to improve spatial and spectral information. It is the best method to solve the problem of skewed

and outliers and send each value to its correct position and group.
- IDA enhances the accuracy of classification models and the training time simultaneously for four different HSI datasets. The code of IDA is available at https://github.com/DalalAL-Alimi/IDA.

2. Improving distribution analysis (IDA)

Besides the HSI difficulties, each dataset has its unique distribution and complexity. Moreover, many FEMs do not care about enhancing data distribution and localization. So, IDA focuses on improving the variance and redistributing each value to its correct location and class. Thus, these processes strengthen choosing only the most robust informative features. In other words, the IDA follows these steps:

1 Decrease the distance between the big and small values, which helps to neglect the outlier values and increase the correlation between values in each class.
2 Enhance the variance between classes and the localization of each value.
3 Improve the data distribution to be more normal and reduce the data complexity.

Let V_{ij} represent the main values of the input dataset, i and j are the dimensions of V. q and g are the numbers of instances and features (bands) respectively, μ is the mean. If we see the following matrix horizontally, we can observe the main values that characterize each class for the other through the full features. Looking vertically at the matrix, we get the features; each feature characterizes the whole band individually.

Different values of each instance

Values of each feature

V_{11}	V_{12}	V_{13}	…	V_{1g}	μ_1
V_{21}	V_{22}	V_{23}	…	V_{2g}	μ_2
V_{31}	V_{32}	V_{33}	…	V_{3g}	μ_3
⋮	⋮	⋮	…	⋮	⋮
V_{q1}	V_{q2}	V_{q3}	…	V_{qg}	μ_q

$$\mu_i = \left[V_{i1}, V_{i2},\ V_{i3}, \ldots,\ V_{iq}\right]^T,\ i = 1, 2, 3, \ldots, q \tag{1}$$

According to the above equation the μ_i is the mean of g instances in ith dimension. So, the first step of IDA is to minimize the gap between the related values, big and small values. Thus the first step of minimizing is calculating the mean (μ) of each instance:

$$\mu_q = \sum_{q=1}^{q} V_q/q \tag{2}$$

The one dimension (1D) of μ_q is converted into 2D, which equals the V dimension, so the new dimension of μ_q will be $q \times g$, where $q \times g = i \times j$ and $\mu_{(q,\ g)} = \mu_q$, as can be seen in the following matrix, each column of $\mu_{(q,\ g)}$ is equal to μ_q, repeating g times:

q

g

μ_{1g}	μ_{1g}	μ_{1g}	…	μ_{1g}
μ_{2g}	μ_{2g}	μ_{2g}	…	μ_{2g}
μ_{3g}	μ_{3g}	μ_{3g}	…	μ_{3g}
⋮	⋮	⋮	…	⋮
μ_{qg}	μ_{qg}	μ_{qg}	…	μ_{qg}

Then, to increase the correlation between the values inside each class, the generated matrix (μ_{qg}) is subtracted from the V_{ij}.

$$S_{qg} = V_{qg} - \mu_{qg} \tag{3}$$

This means that the original features space is transferred into a new space in which values within each class have a more correlated space.

In order to make the values to be smoother and enhancing more the correlation between the related values, the dilation ($\mathcal{D}$) is taken for the results of the third equation and the main values. Dilation works to remove all the intensity fluctuations except the intensity peak according to the main values of V_{ij} and S_{qg}.

$$\mathcal{D}_{qg} = S_{qg} \oplus V_{qg} = \cup_{b \in V} S_b \tag{4}$$

The second step works to improve the variance between classes. To do that, firstly, we subtract the dilation results of Eq. (4) from the main input data again (V_{ij}) to get F_{qg}:

$$F_{qg} = V_{qg} - \mathcal{D}_{qg} \tag{5}$$

So, the calculated values can be represented as follows:

q

g

F_{11}	F_{12}	F_{13}	…	F_{1g}
F_{21}	F_{22}	F_{23}	…	F_{2g}
F_{31}	F_{32}	F_{33}	…	F_{3g}
⋮	⋮	⋮	…	⋮
F_{q1}	F_{q2}	F_{q3}	…	F_{qg}
$\bar{F}_{q1}$	$\bar{F}_{q2}$	$\bar{F}_{q3}$	…	$\bar{F}_{qg}$

Secondly, transferred data is used now to know the main relation between spectral values and increase the differences between them by using the covariance matrix, which can be calculated via the following equation:

$$C_f = \left(\sum_{i=1}^{q} (F_i - \bar{F})(F_i - \bar{F})^T\right)/q - 1 \tag{6}$$

Where $\bar{F}$ is the mean of F, and the dimension of C_f is $d \times d$. C_f calculates the distance observation from the mean of each variable.

In this step, we will focus on the error in measurement. Since the true value of the covariance matrix is not known, the arithmetic

$$C_f = C_{\hat{f}} \mp \varepsilon, \tag{7}$$

where ε is the random error, $C_{\hat{f}}$ is the estimation values.

$\boldsymbol{\varepsilon}$ are small values belonging to [0, 1] and are usually chosen arbitrarily, but the difference here is that we choose $\boldsymbol{\varepsilon}$ depending on the correlation between each class. In other words, $\boldsymbol{\varepsilon}$ is chosen depending on the coefficient correlation. For example, if the coefficient correlation is high, the value of $\boldsymbol{\varepsilon}$ is big, and vice versa. To do that, we generate the error matrix $\boldsymbol{\varepsilon_{qg}}$ depending on the coefficient correlation, then the matrix of covariance C_f is subtracted from the matrix of $\boldsymbol{\varepsilon_{qg}}$, in Eq. (6), to generate a new and more covariance matrix ($G_{d \times d}$).

After the subtraction, k largest eigenvector is selected to compute the eigenvector ($\mathcal{W}$) of $G_{d \times d}$ matrix, where k is the dimensionality of the new feature subspace ($k \leq d$). Now $\mathcal{W}$ represents the weight vector that will be multiplied with $\mathcal{D}$, in Eq. (4), to get the final new dimensional subspace of transferred features as follows:

$$\mathcal{Y}_{q \times k} = \mathcal{D}_{q \times k} \times (\mathcal{W}_{k \times 1})^T \tag{8}$$

Algorithm 1: IDA.

Input: $V \in R^{n\times d}$
Output: $Y \in R^{n\times k}$
1: Reducing the gap by the dilation ($\mathcal{D} = S \oplus V$)
2: Build the covariance matrix ($F.\ \tilde{F}^T$)
3: Correct the results of the covariance matrix ($C \mp \varepsilon$) to increase the correlation.
4: Obtain Eigen and vectors values by linear Eigen decomposition for ($C \mp \varepsilon$)
5: Sort Eigenvalues in decreasing order to sort Eigenvectors
6: Construct matrix $\mathcal{W}$ ($d \times k$) with k top Eigenvectors
7: Transform V to a new subspace Y by using $\mathcal{W}$ and $\mathcal{D}$, ($Y = \mathcal{D}.\mathcal{W}$)

This multiplication operation helps distinguish classes more and chooses the most practical features for classification.

The third and final step is to normalize the data distribution by feeding these new transferred features ($\mathcal{Y}_{q\times k}$) to gaussian transformation:

$$Y = G_Y^{-1}(F_{\mathcal{Y}}(\mathcal{Y})) \tag{9}$$

Where $F_{\mathcal{Y}}$ represents Cumulative Distribution Function (CDF) for the $\mathcal{Y}$, and G_Y^{-1} is the Gaussian Cumulative Distribution Function (GCDF) in y.

3. Experiments

The Python language is used to create and run the improving distribution analysis method (IDA) in a GPU with 26 GB RAM, Windows 10. This study used different hyperspectral datasets to measure the output of IDA and compared it with other related approaches.

3.1. Datasets

The experiments of the study were with four different kinds of datasets. The first one is Indian Pines (IPs) which is developed by AVIRIS sensor and covers 16 classes for Northwest India. The second dataset is Kennedy Space Center (KS_Center), also developed by AVIRIS, and covers 13 classes for KS_Center in Florida; both are available online.[1] The rest two datasets are from HyRANK.[2] The HyRANK dataset provided a large dataset covering several land classes; ISPRS developed it. It includes different hyperspectral images, and this study used Dioni (Di) and Loukia (Lk) [25–27]. The details of all used datasets are in Tables 1 and 2; each color represents a class. Fig. 1 shows the scene of each dataset.

Precision (Pre.) and Recall (Rec.) are used to evaluate the final results and show the accuracy of each class. Precision is used to evaluate the actual positive classification. On the other said, the Recall gives the average of the actual positive computed according to actual positive and negative classification. Kappa accuracy, Overall accuracy, and Average accuracy are used to determine each model's accuracy, as seen in Tables 3-19.

3.2. Comparisons Output of IDA with different FEMs

Having a good idea about the chosen data is a necessary step before initiating any processing. Understanding and visualizing the data give a clear vision of its nature and enable us to do necessary processes to handle any problem and enhance the data before feeding it to the classification model. After visualizing the HSI datasets, it was observed that most of the datasets, if not all of them, have almost the same problems. The pixels or values in the HSI images can be divided into four categories: unique values, shared values, outlier values, and noise values. Fig. 2 simplifies the visualization and the description of these categories. The unique values are the values that differentiate each class from the others and do not exist in the other classes; each class has a particular color and values. The expression "shared values" refers to the fact that several classes share the same values and spectral characteristics in many cases; they can be called mixed or joint values. Shared values are sometimes skewed due to classification errors that hyperspectral imaging systems or sensor failures could cause [28]. Still, upon closer inspection, it can be determined that many of them are primary values for specific classes and contribute to class variation. Outliers are extremist values in a dataset or graph that deviate far from the main pattern of values. However, many outliers may be major values for particular classes and contribute to supporting the variation. The values that do not belong to any class and complicate the classification are called noise values. Due to the spatial context values (pixels) of samples are not fully captured, they can include unreliable pixels because of many other factors like refraction and shadow [1,29,30]. The shared, outlier, and noise values complex FRMs and classification processes.

This study faces to enhance the data distribution and variation and reduce the dimension by obtaining and utilizing only the most informative features. The aims of the first and second steps of IDA are to promote the correlation between small and big related values, reduce the data complexity, and boost localization. In the first step of IDA, reducing the distance between big and small values helps to reduce the quantity of outlier, noise, and mixed values. This step helps to adjust the position of the pixels, precisely choose the most useful features, and accelerate and facilitate the classification process. The histogram in Fig. 3 represents the data distribution of the first band in the IPs dataset, and the boxplot represents the data distribution of the first five bands in the IPs dataset. The histogram in Fig. 3(a) shows that this band has many peaks, the distribution is unstable, and there are different skewed. The boxplot displays the number of outliers and the data distribution in each band; outliers are represented as small circles in the boxplot. As can be seen, there are many outlier values in each band.

We also observe the apparent improvement of IDA output and the differences between the original data distribution in Fig. 3(a) and the distribution data of IDA in Fig. 3(c) in the histogram part. Fig. 3(b) gives a clear view of how the first and second steps enhanced the data and transferred them into a new space by increasing the correlation of the corresponding values and reducing outliers by decreasing the gap between related values before the last step. There is only one peak now, and the data slopes smoothly with almost no outliers in each band. The last process is to normalize the data distribution more and scale it, as shown in Fig. 3(c). The second step is intended to increase the variation between classes. As a result of this separation process, unrelated values (outliers) appear in each band, and Fig. 4 only displays the outliers of the first band in each dataset. In addition, because the normalization process makes the data distribution more normal, the number of outliers also changes. In general, the IDA reduced outliers in the input dataset by more than 80%.

The histograms and boxplots in Fig. 5 represent the data distribution of the first band in the IPs dataset for the other four FEMs (ICA, SVD, UMAP, and PCA). As can be seen, there are many peaks in all compared FEMs. They do not have a specific shape or normal distribution, and there are significant gaps between big and small values. Compared with IDA in Fig. 3, it can be seen the significant differences and how the IDA success in reducing the number of outlier values and improving the distribution, which will play a big role in reducing the training time and increasing the final classification accuracy.

[1] https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
[2] https://www2.isprs.org/commissions/comm3/wg4/hyrank/

Table 1
The four datasets description.

Dataset	Sensor	Band Numbers	Spatial Dimensions	Spatial Resolution	Classes Number
IPs	AVIRIS	200	145×145	20m	16
KS_Center		176	512×614	18m	13
Di	Hyperion sensor (EO-1, USGS)	176	250×1376	30m	12
Lk		176	249×945	30m	14

Table 2
The classes and the samples number in each class for the four datasets.

#	IPs Dataset		KS Center Dataset		HyRANK Datasets		
	Classes	Samples	Classes	Samples	Classes	Dioni Samples	Loukia Samples
1	Alfalfa	46	Scrub	761	Dense Urban Fabric (DUF)	1262	288
2	Corn-notill (CN)	1428	Willow swamp (WS)	243	Mineral Extraction Sites (MES)	204	67
3	Corn-mintill (CM)	830	Cabbage palm hammock (CPH)	256	Non Irrigated Arable Land (NIAL)	614	542
4	Corn	237	Cabbage palm/oak hammock (CPOH)	252	Fruit Trees (FT)	150	79
5	Grass-pasture (GP)	483	Slash pine (SP)	161	Olive Groves (OG)	1768	1401
6	Grass-trees (GT)	730	Oak/broadleaf hammock (OBH)	229	Broad-Leaved Forest (BLF)	–	223
7	Grass-pasture-mowed (GPM)	28	Hardwood swamp (HS)	105	Coniferous Forest (CF)	361	500
8	Hay-windrowed (HW)	478	Graminoid marsh (GM)	431	Mixed Forest (MF)	–	1072
9	Oats	20	Spartina marsh (SM)	520	Dense Sclerophyllous Vegetation (DSV)	5035	3793
10	Soybean-notill (SN)	972	Cattail marsh (CM)	404	Sparce Sclerophyllous Vegetation (SSV)	6374	2803
11	Soybean-mintill (SM)	2455	Salt marsh (SM)	419	Sparcely Vegetated Areas (SVA)	1754	404
12	Soybean-clean (SC)	593	Mud flats (MF)	503	Rocks and Sand (RS)	492	487
13	Wheat	205	Water	927	Water	1612	1393
14	Woods	1265			Coastal Water (CW)	398	451
15	Buildings-Grass-Trees-Drives (BGTD)	386					
16	Stone-Steel-Towers (SST)	93					

3.3. Evaluation Results with different FEMs

In this paper, the classifier models are divided into machine learning methods, including SVM and multilayer perceptron (NN) [28], and deep learning methods, which are 2D-CNN [28] and spectral-spatial feature extraction methods. SVM and NN work to extract the spectral features. 2D-CNN obtains the spatial feature, while spectral-spatial feature extraction methods are used to extract the spectral-spatial features simultaneously, like HybridCNN [31] and HybridSN [32], which were used in this study [18,28].

In order to evaluate the performance of the proposed method, firstly, IDA was compared with four different well-known feature extraction reduction methods: ICA, SVD, UMAP, and PCA [3,15,33]. They were compared with three classifier models and four different datasets, and each dataset was split into 20% and 80% as training and testing sets. The number of components (features) for all these preprocessing methods is 15 for all datasets. This subsection uses the three models, SVM, NN, and 2D-CNN. The NN is one layer of fully connected (FC). The 2D-CNN model includes two layers of 2D-CNN, (5,5) Kernel-Size, one Max Pooling (2, 2), and one FC with 100 units. NN and 2D-CNN use Relu as an activation function and 15 window size. The number of epochs is 100 in all models. All used FEMs reduced the number of bands to 15 bands.

Tables 1 and 2 illustrate how diverse each dataset is. All of them have distinct dimensions, resolutions, class numbers, and samples. IDA's testing and evaluation are significantly complicated by these variances. Moreover, each dataset has its own data distribution, providing an excellent opportunity to evaluate the sug-

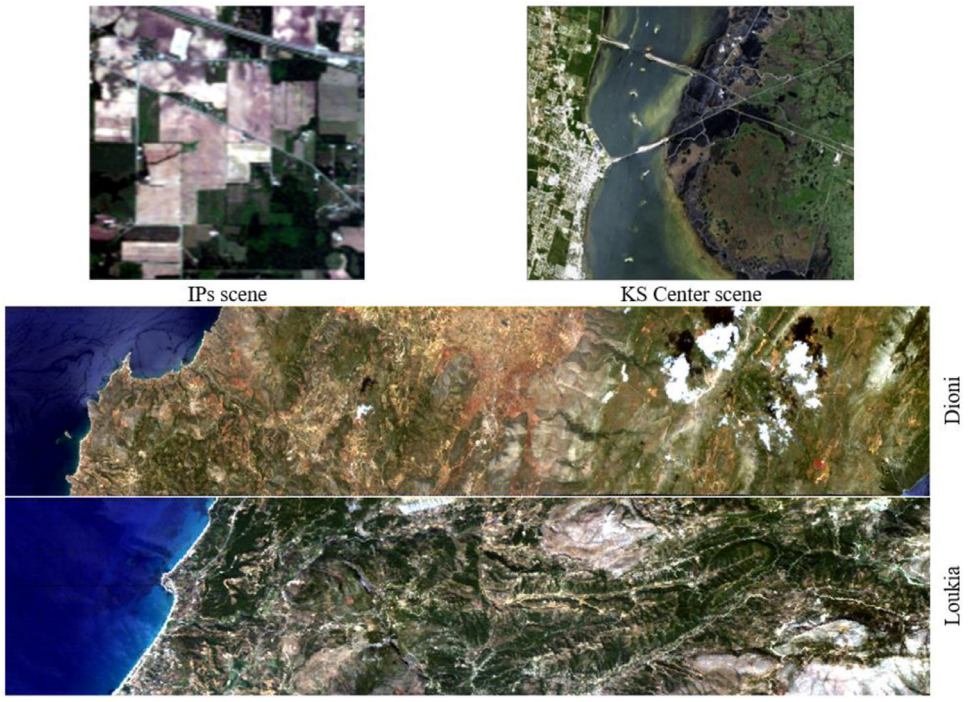

Fig. 1. The four datasets' scenes.

Table 3
The results of SVM of the different preprocessing models for IPs dataset.

No	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Alfalfa	37	77	46	0	0	0	0	77	46	82	76
2	CN	1143	70	72	53	49	47	48	70	72	68	76
3	CM	664	66	68	70	23	83	33	66	68	70	74
4	Corn	190	62	54	73	17	73	13	62	54	57	43
5	GP	386	86	89	87	66	85	71	86	89	85	89
6	GT	584	88	96	82	96	72	96	88	96	95	93
7	GPM	22	81	95	0	0	0	0	81	95	48	55
8	HW	382	95	99	86	100	82	99	95	99	97	96
9	Oats	16	57	50	0	0	0	0	57	50	47	50
10	SN	778	73	68	49	34	56	44	73	68	68	66
11	SM	1964	79	82	51	86	53	84	79	82	79	81
12	SC	475	74	67	50	3	70	7	74	67	77	65
13	Wheat	164	85	95	84	93	87	98	85	95	96	99
14	Woods	1012	93	90	84	98	86	97	93	90	96	92
15	BGTD	309	55	46	73	35	70	17	55	46	80	75
16	SST	74	97	84	100	84	98	84	97	84	96	74
Kappa accuracy (%)			75.35		57.26		58.12		75.34		76.87	
Overall accuracy (%)			78.43		63.78		64.35		78.41		79.76	
Average accuracy (%)			75.05		49.00		49.32		75.04		75.33	
Training Time (s)			0.33		0.19		0.21		0.25		0.24	
Test Time (s)			1.39		1.30		1.05		1.05		1.05	

Table 4
The results of NN of the different preprocessing models for IPs dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Alfalfa	37	0	0	0	0	0	0	17	41	0	0
2	CN	1143	0	0	41	34	55	34	46	56	52	45
3	CM	664	0	0	6	1	2	0	42	31	46	24
4	Corn	190	0	0	30	4	94	8	26	30	0	0
5	GP	386	0	0	29	34	46	73	61	67	58	56
6	GT	584	0	0	83	78	67	84	86	86	76	87
7	GPM	22	0	0	0	0	0	0	59	74	0	0
8	HW	382	0	0	85	90	57	76	92	90	81	95
9	Oats	16	0	0	0	0	0	0	20	6	0	0
10	SN	778	0	0	45	4	0	0	45	33	53	47
11	SM	1964	27	96	44	88	42	91	57	61	55	85
12	SC	475	0	0	0	0	32	9	38	32	34	13
13	Wheat	164	0	0	46	59	72	90	79	76	89	95
14	Woods	1012	45	55	73	68	67	60	87	79	75	90
15	BGTD	309	0	0	21	34	0	0	39	43	53	13
16	SST	74	0	0	98	77	0	0	57	81	74	19
Kappa accuracy (%)			9.80		39.60		39.66		52.19		53.87	
Overall accuracy (%)			29.78		49.24		49.62		58.18		60.68	
Average accuracy (%)			9.44		35.62		32.93		55.35		41.71	
Training Time (s)			10.03		14.82		9.84		13.00		13.49	
Test Time (s)			0.31		0.40		0.30		0.37		0.39	

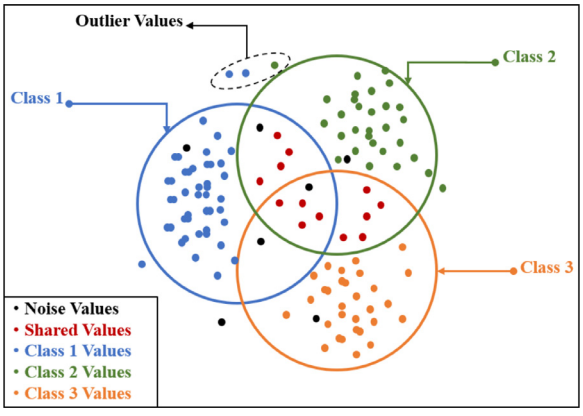

Fig. 2. A small example illustrates the visualization of data distribution.

gested method with greater complexity. The outcomes of the three classifier models are shown in Tables 3-14 for the four datasets.

In this study, the IPs dataset includes the biggest number of classes, 16 classes. In each class, the number of samples is dissimilar. Some classes have very small samples compared to the big number in others; 20–2455 samples, in Table 2. Moreover, the data distribution is very complex. Because of all of these complexities, not all FEMs can handle them. On the other hand, the IDA method produces exceptional accuracy in the three classification models, as can be seen in Tables 3-5.

The KS_Center is also a multi-class dataset, which has 13 classes. The number of samples of each class is better than the number in the IPs dataset, 105–927 samples, in Table 2. By contrast, the data distribution is more complex than in the IPs dataset. For that reason, it was the most complex dataset to get very high accuracy. Significantly, the ICA is the worst FEM to extract the features and enhance the accuracy in all classification models. With the KS_Center dataset, IDA provided a very high accuracy. It was the best with the 2D-CNN model and the second-best method after UMAP with the SVM and NN models, as seen in Tables 6-8.

The Di and Lk datasets have the least complex data distribution, which is linear. In the Di dataset, the sample number in each class is very high, but the outlier values are huge, as detailed in Table 2 and Fig. 4. Some classes have a small number of samples in the Lk dataset, like the situation in the IPs dataset, and the number of outliers is very large. Tables 9 and 12 demonstrate that both ICA and PCA achieved the same accuracy, but PCA prevailed in terms of performance time; ICA excels with spatial data. The IDA provides good performance with the Di dataset, Tables 9-11. With SVM and NN, the IDA was the third-best preprocessing enhancing spectral information in the Lk dataset.

As can be observed, the outcomes are dependent not just on the FEM but also on the classification model employed. The accuracy of the used FEMs is different in each classifier model. For example, ICA with SVM and 2D-CNN gives very high accuracy. On the other hand, ICA with NN gives very low accuracy. In general, the SVM is affected by the feature reduction methods, while NN is affected more by the number of training samples; NN is better when it has enough training samples.

It is evident from the IPs dataset results that the IDA results are the best. 2D-CNN achieves the maximum accuracy for all FEMs

Table 5
The results of 2D-CNN of the different preprocessing models for IPs dataset.

No	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Alfalfa	37	97	89	82	84	100	73	56	62	100	100
2	CN	1143	91	92	58	59	84	66	64	57	99	99
3	CM	664	98	93	64	57	78	77	61	43	98	100
4	Corn	190	79	95	59	49	86	91	53	25	98	100
5	GP	386	93	96	88	87	98	95	85	91	100	97
6	GT	584	98	99	93	99	88	93	84	96	100	99
7	GPM	22	100	78	58	48	100	96	25	17	100	100
8	HW	382	100	99	98	98	97	100	89	97	100	100
9	Oats	16	100	31	50	31	53	63	3	6	100	100
10	SN	778	95	88	51	34	67	81	74	39	99	100
11	SM	1964	96	98	65	79	84	87	66	86	100	99
12	SC	475	92	93	67	53	70	67	51	59	99	98
13	Wheat	164	100	100	94	90	94	99	84	93	100	100
14	Woods	1012	96	99	98	98	96	100	94	97	100	100
15	BGTD	309	93	90	89	86	93	73	80	56	99	99
16	SST	74	100	91	96	88	80	86	50	59	96	97
Kappa accuracy (%)			94.17		69.34		82.22		67.79		99.15	
Overall accuracy (%)			94.89		73.37		84.41		72.08		99.26	
Average accuracy (%)			89.42		71.26		84.21		61.46		99.28	
Training Time (s)			30.74		29.67		30.12		30.50		29.44	
Test Time (s)			0.70		0.58		0.69		0.69		0.67	

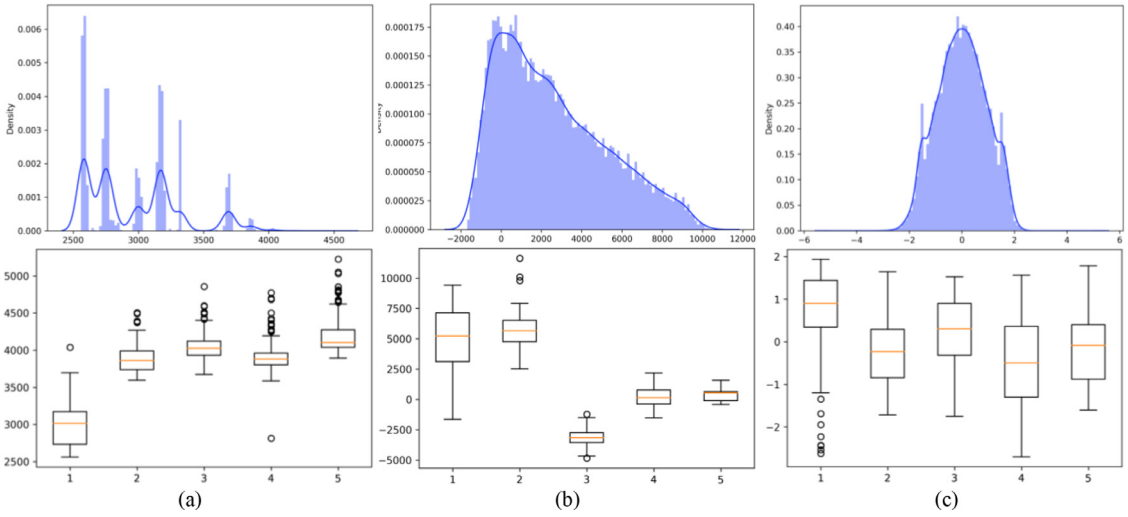

Fig. 3. (a) represents the main data distribution for the first band of the IPs dataset. (b) The results of Eq. (8) represent the second step of IDA feature reduction. (c) is the data distribution after normalization, the final output of improving distribution analysis.

across all various datasets. As demonstrated in Tables 6 and 7, UMAP with SVM and NN provides the maximum accuracy for the KS CENTER dataset. PCA with NN yields the highest precision for the Di and Lk datasets. Alternatively, IDA maintains a balance and gives high accuracy with all classifier models and datasets.

The IDA shows the differences and exceptional accuracy with 2D-CNN in all datasets. The IDA corrected the positions of the pixels, which enhanced localization. It improved spatial information in each band (feature) and decreased noise and mixed values. The 2DCNN classification approach is more effective at extracting spatial than spectral information. Consequently, the IDA provided the 2DCNN with more precise spatial data; thus, the 2DCNN produced better results than other models. For instance, IDA accuracy in the IPs dataset is 30% greater than SVD and PCA. Additionally, it is 36% greater than SVD and 41% greater than PCA for the KS CENTER dataset. IDA was 3% more accurate in the Di dataset than ICA

Table 6
The results of SVM of the different preprocessing models for the KS_CENTER dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Scrub	609	30	97	40	98	92	95	36	96	65	82
2	WS	194	75	2	0	0	82	80	0	0	26	34
3	CPH	205	0	0	0	0	58	79	0	0	60	64
4	CPOH	202	0	0	0	0	54	45	0	0	40	32
5	SP	129	0	0	0	0	49	38	0	0	25	19
6	OBH	183	0	0	0	0	60	48	0	0	32	24
7	HS	84	0	0	0	0	74	70	0	0	43	32
8	GM	345	22	25	19	15	78	74	27	23	50	55
9	SM	416	0	0	63	88	80	92	57	64	76	76
10	CM	323	60	2	60	2	93	85	60	2	49	43
11	SM	335	100	86	98	88	90	96	100	86	97	87
12	MF	402	50	55	46	54	93	86	49	62	59	50
13	Water	742	72	99	71	99	100	100	77	100	90	91
Kappa accuracy (%)			37.77		47.50		82.3		45.77		59.87	
Overall accuracy (%)			46.34		54.47		84.14		52.99		64.09	
Average accuracy (%)			28.11		34.21		76.09		33.19		53.05	
Training Time (s)			0.06		0.08		0.03		0.08		0.09	
Test Time (s)			0.25		0.35		0.16		0.37		0.29	

Table 7
The results of the NN of the different preprocessing models for the KS_CENTER dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Scrub	609	0	0	42	68	75	95	43	57	43	69
2	WS	194	0	0	33	31	52	33	12	8	0	0
3	CPH	205	0	0	0	0	7	3	23	36	12	3
4	CPOH	202	0	0	20	8	21	57	0	0	29	18
5	SP	129	0	0	2	2	0	0	3	5	0	0
6	OBH	183	0	0	3	5	40	9	1	1	0	0
7	HS	84	0	0	25	27	0	0	0	0	0	0
8	GM	345	0	0	6	5	31	31	19	20	40	26
9	SM	416	0	0	37	61	60	63	11	5	41	55
10	CM	323	0	0	14	5	72	44	20	14	44	21
11	SM	335	0	0	36	57	85	94	90	90	42	85
12	MF	402	100	0	16	6	63	84	31	21	63	32
13	Water	742	18	100	92	73	99	98	74	96	61	99
Kappa accuracy (%)			0.04		30.27		59.70		32.88		40.21	
Overall accuracy (%)			17.78		37.67		64.04		40.16		47.72	
Average accuracy (%)			7.71		26.83		47.00		27.12		31.28	
Training Time (s)			4.24		5.71		7.13		6.68		8.16	
Test Time (s)			0.16		0.20		0.19		0.19		0.25	

Table 8
The results of 2D-CNN of the different preprocessing models for the KS_CENTER dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	Scrub	609	46	96	87	92	100	100	79	90	100	100
2	WS	194	42	13	43	35	98	97	35	26	99	99
3	CPH	205	0	0	68	37	98	98	44	68	99	99
4	CPOH	202	92	24	41	46	95	89	32	20	100	99
5	SP	129	85	35	63	57	87	89	51	49	100	100
6	OBH	183	100	2	48	54	95	97	32	8	100	100
7	HS	84	0	0	47	60	100	100	44	8	100	100
8	GM	345	32	83	53	73	96	99	48	77	99	100
9	SM	416	0	0	87	75	99	98	90	66	100	100
10	CM	323	53	43	51	46	100	100	53	44	99	98
11	SM	335	97	92	81	72	100	99	84	69	100	100
12	MF	402	67	68	55	57	99	100	52	57	99	100
13	Water	742	87	100	78	80	100	100	71	82	100	100
Kappa accuracy (%)			53.22		63.24		98.21		58.28		99.52	
Overall accuracy (%)			58.85		67.03		98.39		62.79		99.57	
Average accuracy (%)			42.75		60.25		97.41		51.18		99.47	
Training Time (s)			27.75		24.49		19.26		22.89		20.22	
Test Time (s)			0.33		0.40		0.33		0.41		0.33	

Table 9
The results of SVM of the different preprocessing models for the Dioni dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	1010	83	85	85	79	80	63	83	85	90	85
2	MES	163	98	83	98	77	94	81	98	83	94	82
3	NIAL	491	89	87	90	89	83	87	89	87	90	89
4	FT	120	88	69	92	71	92	61	88	69	89	83
5	OG	1414	90	92	90	92	90	87	90	92	92	90
6	CF	289	100	100	97	100	92	100	100	100	95	98
7	DSV	4028	94	92	95	90	95	89	94	92	96	94
8	SSV	5099	92	94	89	95	87	95	92	94	94	96
9	SVA	1403	93	91	87	84	81	89	93	91	93	95
10	RS	394	98	95	99	92	99	89	98	95	99	98
11	W	1290	100	100	99	100	98	100	100	100	94	97
12	CW	318	100	99	100	94	99	91	100	99	86	77
Kappa accuracy (%)			91.13		89.32		87.14		91.13		92.09	
Overall accuracy (%)			92.85		91.42		89.68		92.85		93.63	
Average accuracy (%)			90.70		88.60		85.86		90.70		90.35	
Training Time (s)			0.46		0.24		0.25		0.44		0.33	
Test Time (s)			2.28		1.92		1.56		2.03		2.09	

and PCA. In addition, IDA is 21% superior to PCA, 17% superior to SVD, 7% greater than UMAP, and 7.5% greater than ICA in the Lk dataset, according to Tables (8,11,14). Figs. 6-9 show the classification results of the output of the various feature extraction methods by the 2D-CNN model for the four datasets. Because the 2D-CNN is the most accurate model for all FRMs and all datasets, Fig. 10 depicts the average accuracy of each FEM with 2D-CNN during training; as shown, the IDA has the highest and most steady accuracy. The IDA generally offers the highest precision because it improves data distribution and decreases data complexity and outliers.

Table 10
The results of the NN of the different preprocessing models for the Dioni dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	1010	0	0	62	56	55	57	85	73	77	68
2	MES	163	0	0	70	80	0	0	50	65	83	44
3	NIAL	491	0	0	79	78	0	0	67	81	79	76
4	FT	120	0	0	30	37	0	0	68	65	75	25
5	OG	1414	0	0	82	80	62	50	83	90	78	78
6	CF	289	0	0	74	81	97	74	94	87	97	74
7	DSV	4028	82	0	89	80	88	90	90	89	88	91
8	SSV	5099	32	100	78	91	78	89	86	86	86	89
9	SVA	1403	0	0	70	49	59	73	80	76	81	80
10	RS	394	0	0	79	87	98	88	88	95	93	95
11	W	1290	0	0	99	99	82	100	99	100	85	99
12	CW	318	0	0	97	96	100	10	90	94	85	29
Kappa accuracy (%)			0.11		75.98		70.97		83.03		80.85	
Overall accuracy (%)			31.90		80.80		77.02		86.28		84.67	
Average accuracy (%)			8.36		76.15		52.62		83.32		70.72	
Training Time (s)			14.20		13.42		19.7		15.58		20.72	
Test Time (s)			0.56		0.73		0.66		0.51		0.67	

Table 11
The results of 2D-CNN of the different preprocessing models for the Dioni dataset.

No.	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	1010	97	91	91	84	84	82	94	89	99	98
2	MES	163	100	98	91	96	97	98	99	91	100	98
3	NIAL	491	93	94	91	86	89	78	91	87	98	98
4	FT	120	100	33	70	59	92	61	84	65	96	92
5	OG	1414	93	99	93	95	92	94	94	99	99	99
6	CF	289	99	100	97	94	100	96	100	100	100	100
7	DSV	4028	98	94	97	97	98	95	97	99	100	100
8	SSV	5099	94	99	95	96	94	97	98	96	100	100
9	SVA	1403	98	97	93	91	91	90	94	98	100	100
10	RS	394	100	96	90	97	82	99	99	98	100	100
11	W	1290	100	100	100	100	100	100	100	100	100	100
12	CW	318	100	100	100	100	100	100	100	100	100	100
Kappa accuracy (%)			95.49		93.75		92.74		96.06		99.44	
Overall accuracy (%)			96.37		94.97		94.15		96.82		99.54	
Average accuracy (%)			91.72		91.33		90.71		93.55		98.62	
Training Time (s)			53.53		42.98		40.06		48.36		45.40	
Test Time (s)			1.16		0.97		0.94		0.97		1.11	

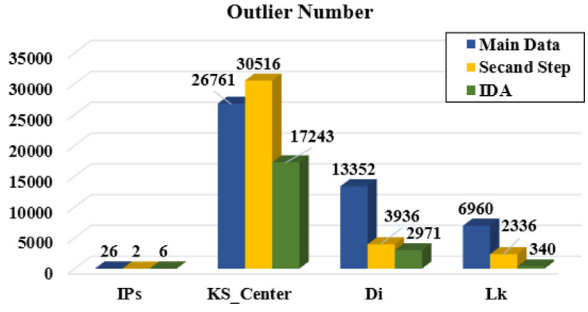

Fig. 4. The number of outliers of the first band in each dataset, the number after the second step of improving distribution analysis, and the outliers number in the final results for the IDA for the first band.

3.4. Spectral-spatial extraction and time performance

Using 2D-CNN and 3D-CNN is a very efficient way to extract spectral-spatial information from the HSI in deep learning algorithms [34–37]. Firstly, processing spectral-spatial features need to create the input data cube before feeding it into 3D-CNN. The 3D-CNN layers extract spectral-spatial feature maps, while the 2D-CNN layers boost spatial features [38,39]. In this subsection, IDA is used as FRM for HybridCNN [31], HybridSN [32], and FUSENet [40] models and compared with their original FRM, which is PCA.

Tables 15 and 16 show that the IDA significantly impacts classification accuracy for all classifier models. Although the results of the Di and Lk datasets are almost identical for PCA and IDA, IDA substantially decreases training time, as seen in Tables 17 and 18. Adding 2D-CNN after 3D-CNN layers reduces the complexity of

Table 12
The results of SVM of the different preprocessing models for the Loukia dataset.

No	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	144	81	73	64	55	60	47	81	73	71	63
2	MES	33	100	81	98	83	92	85	100	81	100	87
3	NIAL	271	91	89	87	83	66	73	91	89	81	79
4	FT	39	64	70	90	14	0	0	64	70	42	38
5	OG	701	89	92	88	90	58	49	89	92	68	72
6	BLF	111	63	69	72	54	68	47	63	69	65	58
7	CF	250	63	70	80	56	71	53	63	70	62	65
8	MF	536	63	57	66	56	61	51	63	57	67	63
9	DSV	1897	78	77	75	84	67	79	78	77	76	75
10	SSV	1402	82	85	80	85	75	78	82	85	78	81
11	SVA	202	92	79	87	45	74	39	92	79	86	68
12	RS	244	95	97	89	90	83	92	95	97	92	94
13	W	697	100	100	100	100	100	100	100	100	99	99
14	CW	225	100	100	100	100	100	100	100	100	96	96
Kappa accuracy (%)			79.22		77.58		67.17		79.22		73.71	
Overall accuracy (%)			82.50		81.34		72.73		82.50		77.88	
Average accuracy (%)			81.25		71.15		63.79		81.25		74.20	
Training Time (s)			0.36		0.22		0.25		0.35		0.34	
Test Time (s)			1.60		1.40		1.45		1.50		1.57	

Table 13
The results of the NN of the different preprocessing models for the Loukia dataset.

No	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	144	0	0	46	11	0	0	60	50	58	21
2	MES	33	0	0	30	59	0	0	48	50	0	0
3	NIAL	271	0	0	55	61	45	47	72	74	71	63
4	FT	39	0	0	58	22	0	0	52	46	100	3
5	OG	701	0	0	80	82	43	12	84	82	64	54
6	BLF	111	0	0	0	0	68	29	36	48	70	31
7	CF	250	0	0	21	5	63	48	53	56	68	43
8	MF	536	0	0	56	56	54	50	54	46	60	55
9	DSV	1897	28	100	67	76	60	78	70	68	66	77
10	SSV	1402	0	0	68	78	61	80	74	77	69	78
11	SVA	202	0	0	39	23	78	17	57	50	59	50
12	RS	244	0	0	74	68	69	77	67	91	82	82
13	W	697	0	0	100	100	100	100	100	100	87	99
14	CW	225	0	0	99	100	100	100	95	100	91	59
Kappa accuracy (%)			0.00		64.57		56.70		68.16		63.35	
Overall accuracy (%)			28.09		70.66		64.67		73.10		69.72	
Average accuracy (%)			7.14		53.03		45.47		67.00		51.17	
Training Time (s)			8.16		8.23		11.32		11.83		15.17	
Test Time (s)			0.31		0.29		0.36		0.37		0.46	

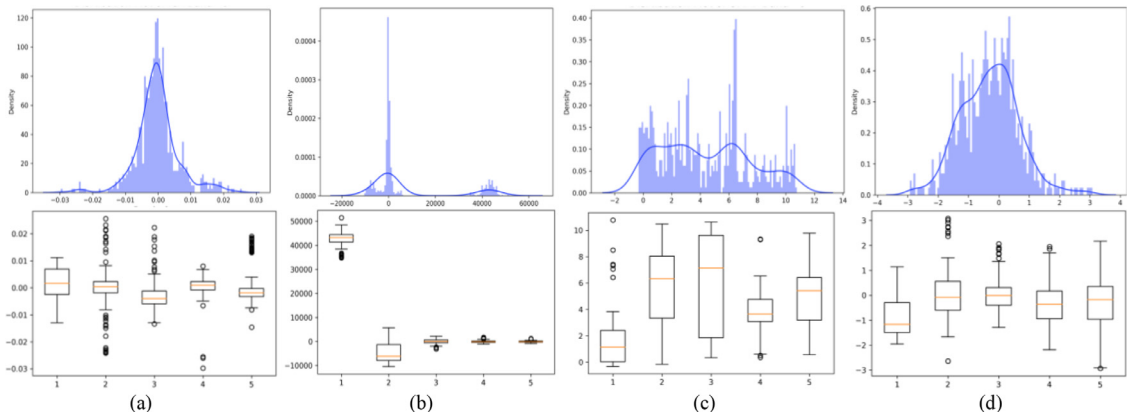

Fig. 5. (a) represents the output of ICA data distribution. (b) is the SVD output. The obtained feature of UMAP is represented in (c). (d) is the representation of first band data of the PCA output. The four examples are for the first band of the IPs dataset.

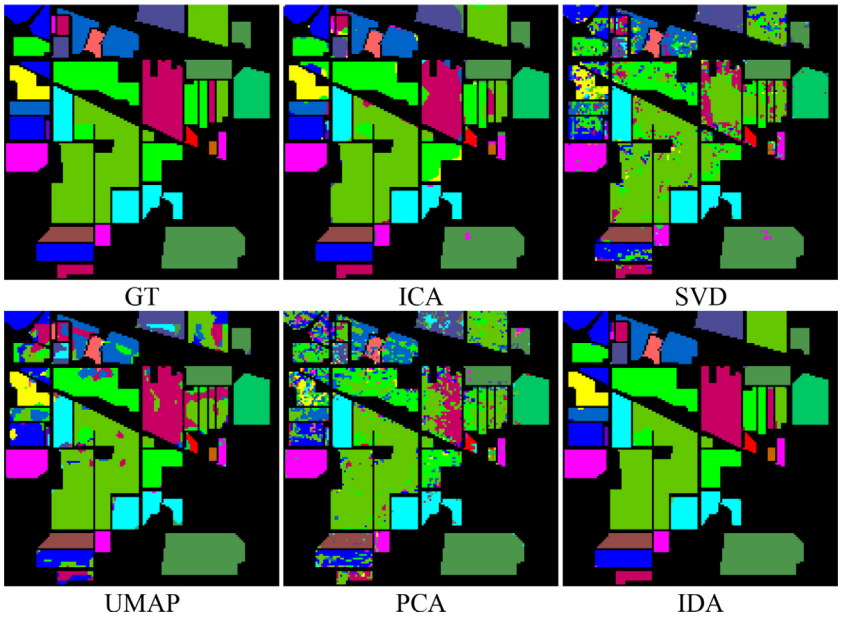

Fig. 6. The output of 2D-CNN Of IPs dataset for each FEMs and the first image is the first image is the ground-truth (GT) image.

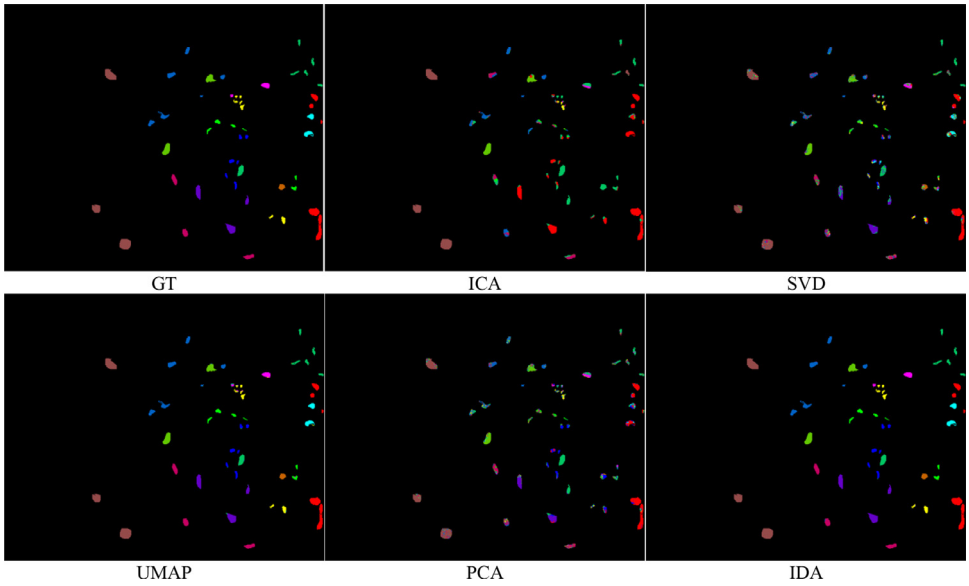

Fig. 7. The output of the 2D-CNN Of KS_CENTER dataset for each FEMs and the first image is the first image is the ground-truth (GT) image.

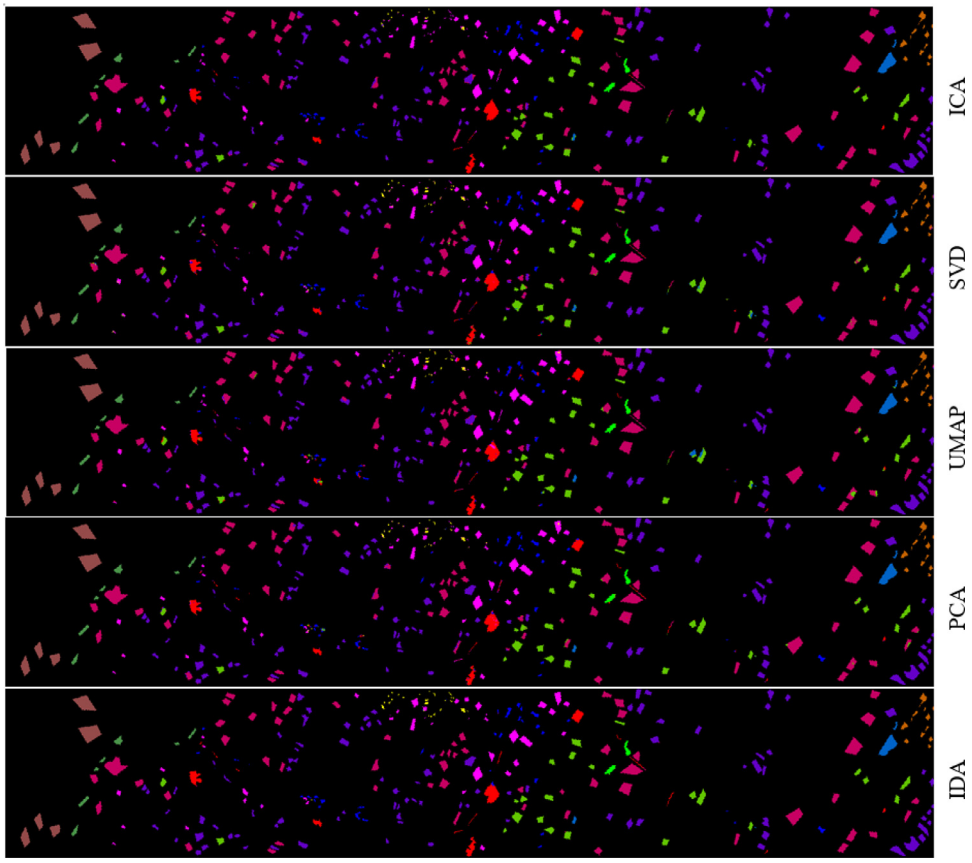

Fig. 8. The output of the 2D-CNN Of Dioni dataset for each FEMs.

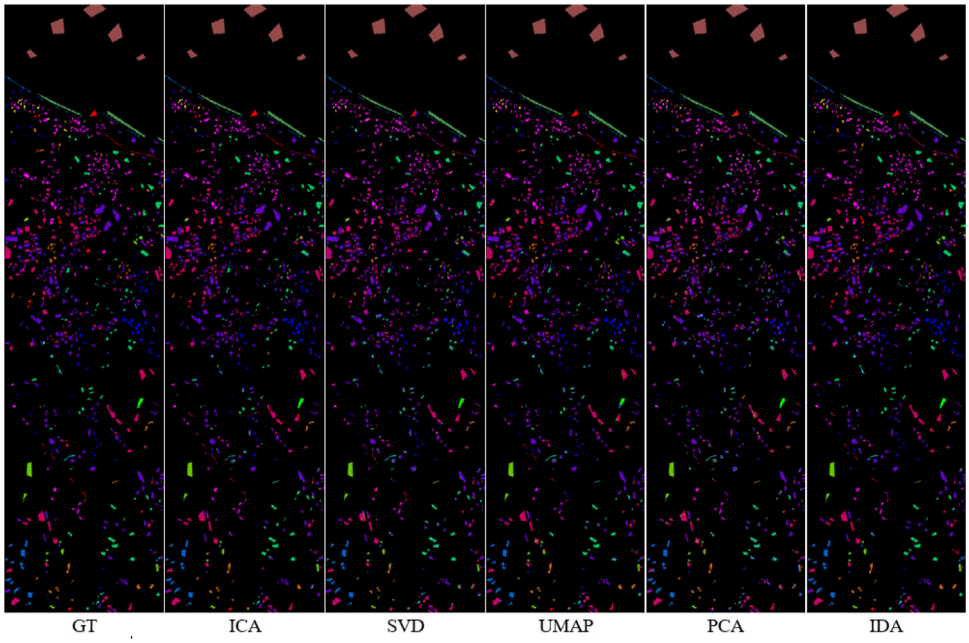

Fig. 9. The output of the 2D-CNN Of Loukia dataset for each FEMs.

3D and enhances object localization (spatial features). The FUSENet model obtains less accuracy than HybridCNN and HybridSN since it does not include 2D-CNN, Tables 15-18. In general, the IDA pre-processing introduced the best spectral-spatial extraction for the three classification models utilized. Moreover, it is the quickest in terms of training and testing time, indicating that IDA significantly enhanced the reduced data and its complexity prior to feeding it to the classifier model.

It is evident from all comparisons that the IDA enhanced the extraction of the spectral features, spatial features, and spectral-spatial, especially the spatial features extraction. Table 19 shows the performance accuracy of IDA with three different numbers of

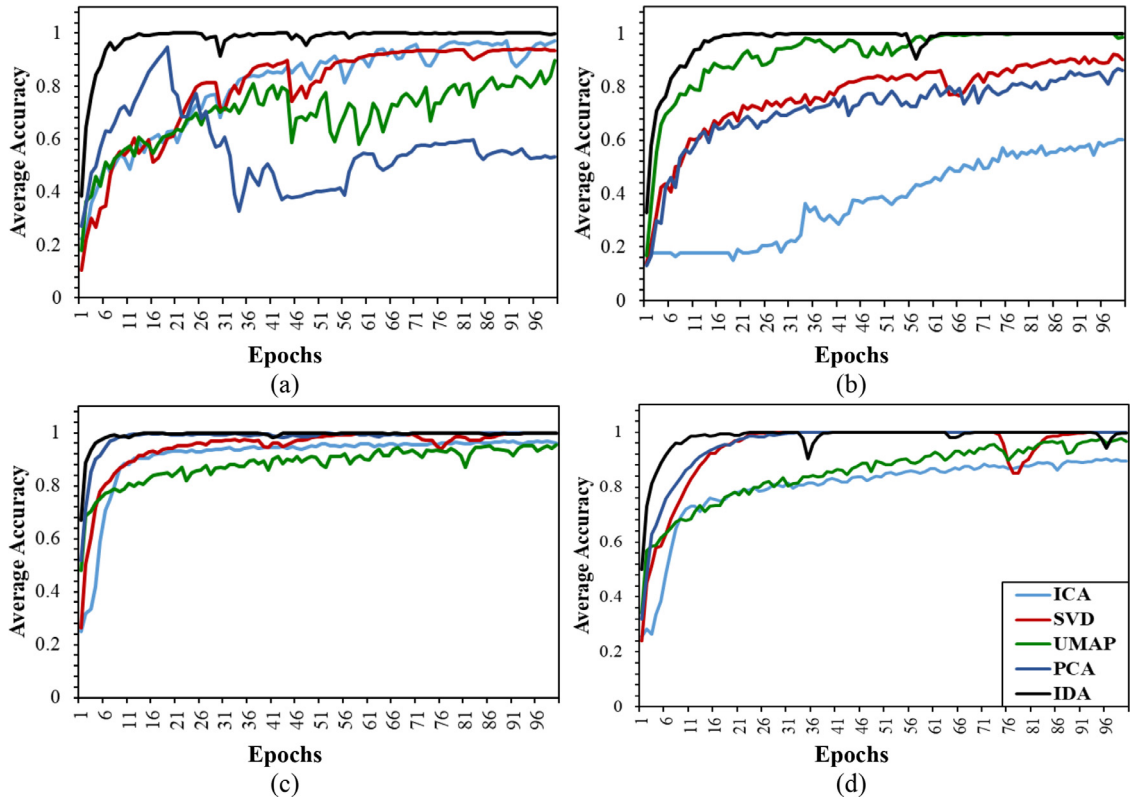

Fig. 10. The average accuracy of each FEM obtained by 2D-CNN for all datasets: (a) IPs, (b) KS_CENTER, (c) Dioni, and (d) Loukia datasets.

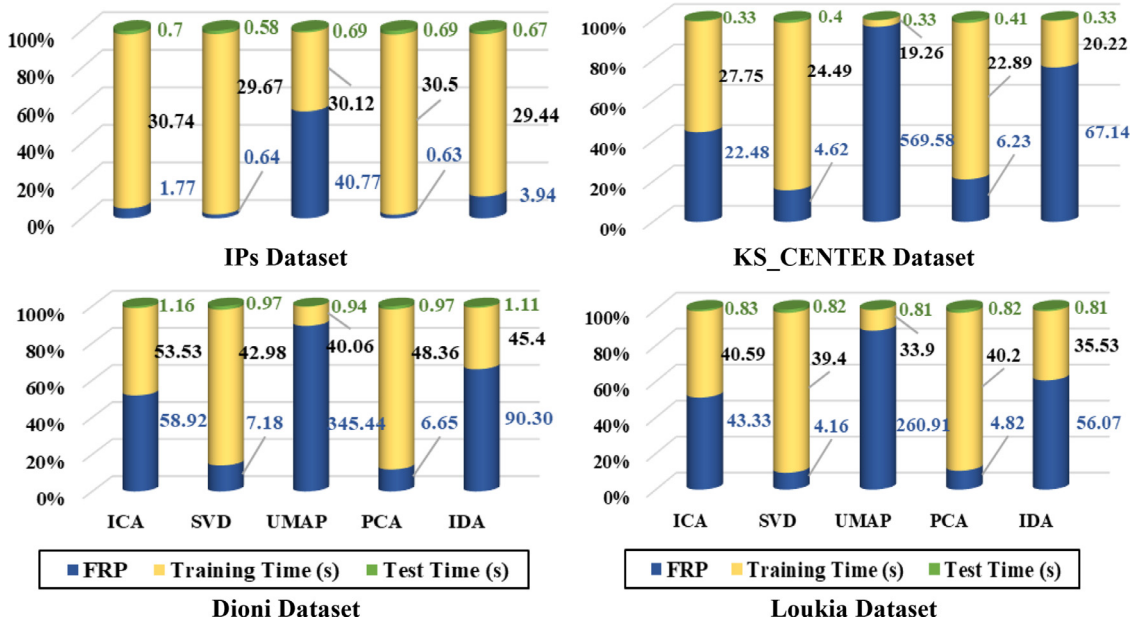

Fig. 11. The comparison of performance time for the feature reduction process (FRP), training, and testing in 2D-CNN for the four used datasets.

components (10, 15, 30); the number of used bands. Due to the structure of HybridSN does not accept the ten bands of transferred features, it was trained with (15, 30) bands, as seen in Table 19. IDA introduced a successful performance with all the different numbers of components in all used classifier models.

When talking about feature reduction or preprocessing, it must be taken into consideration the amount of time taken by that process. So, Fig. 11 summarizes the four datasets' preprocessing, training, and testing time in 2D-CNN. It shows the differences in the processing time that each FEM takes to reduce the input data dimension. The UMAP takes longer in the feature reduction process (FRP) than others. Although ICA, SVD, and PCA introduce shorter times in FRP, they give lower accuracy and longer training time than IDA and, in some cases, longer testing time than IDA. Generally, besides the highest accuracy of IDA, it provides a good balance between FRP and training time, as evidenced in Tables 3-14 and Fig. 11. Furthermore, IDA accelerates the process of spectral-spatial extraction models (Tables 15-18).

Table 14
The results of 2D-CNN of the different preprocessing models for the Loukia dataset.

No	Classes (%)	Testing Samples	ICA		SVD		UMAP		PCA		IDA	
			Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1	DUF	144	88	90	59	44	82	73	51	43	94	84
2	MES	33	100	94	94	89	100	67	60	91	100	100
3	NIAL	271	99	85	75	72	87	87	69	66	96	94
4	FT	39	88	35	21	10	79	43	11	13	76	71
5	OG	701	89	98	81	87	83	83	84	80	92	95
6	BLF	111	99	40	56	53	83	58	30	28	80	69
7	CF	250	86	58	63	51	90	77	61	59	92	88
8	MF	536	73	84	70	68	83	83	73	67	95	94
9	DSV	1897	84	86	76	79	85	90	76	75	92	92
10	SSV	1402	86	88	79	79	90	89	75	77	92	95
11	SVA	202	96	81	77	77	85	89	55	67	97	98
12	RS	244	94	94	87	92	92	97	88	90	97	99
13	W	697	100	100	100	100	100	100	99	100	100	100
14	CW	225	100	100	98	100	99	100	98	100	100	100
Kappa accuracy (%)			85.22		75.42		85.89		72.07		92.71	
Overall accuracy (%)			87.60		79.35		88.15		76.45		93.86	
Average accuracy (%)			81.01		71.46		81.10		68.31		91.40	
Training Time (s)			40.59		39.40		33.9		40.20		35.53	
Test Time (s)			0.83		0.82		0.81		0.82		0.81	

Table 15
The three models to classify the spectral-spatial features for the IPs dataset.

Performance Evaluation	PCA-HybridCNN	IDA-HybridCNN	PCA-HybridSN	IDA-HybridSN	PCA-FUSENet	IDA-FUSENet
Kappa accuracy (%)	98.41	99.59	98.39	98.37	62.24	77.70
Overall accuracy (%)	98.61	99.64	98.59	98.57	66.78	80.38
Average accuracy (%)	97.00	99.12	96.30	98.84	48.74	63.54
Training Time (s)	83.00	59.40	22.07	17.66	4344.71	3438.02
Test Time (s)	1.60	1.83	0.71	0.74	77.60	60.02

Table 16
The three models to classify the spectral-spatial features for the KS_CENTER dataset.

Performance Evaluation	PCA-HybridCNN	IDA-HybridCNN	PCA-HybridSN	IDA-HybridSN	PCA-FUSENet	IDA-FUSENet
Kappa accuracy (%)	85.98	98.84	86.90	98.80	55.05	96.12
Overall accuracy (%)	87.45	98.96	88.25	98.92	60.64	96.52
Average accuracy (%)	84.35	98.42	83.09	98.42	68.67	96.83
Training Time (s)	83.12	35.24	41.81	8.74	2006.90	1564.82
Test Time (s)	0.84	0.92	0.37	0.41	39.39	30.35

Table 17
The three models to classify the spectral-spatial features for the Dioni dataset.

Performance Evaluation	PCA-HybridCNN	IDA-HybridCNN	PCA-HybridSN	IDA-HybridSN	PCA-FUSENet	IDA-FUSENet
Kappa accuracy (%)	99.11	99.35	99.13	99.24	98.72	98.20
Overall accuracy (%)	99.28	99.48	99.30	99.39	98.97	98.54
Average accuracy (%)	98.03	98.74	97.94	98.19	98.74	97.94
Training Time (s)	95.33	94.71	48.06	33.72	6527.21	5966.79
Test Time (s)	3.38	3.40	1.30	1.27	117.37	115.17

Table 18
The three models to classify the spectral-spatial features for the Loukia dataset.

Performance Evaluation	PCA-HybridCNN	IDA-HybridCNN	PCA-HybridSN	IDA-HybridSN	PCA-FUSENet	IDA-FUSENet
Kappa accuracy (%)	91.88	94.25	92.90	92.42	90.82	88.93
Overall accuracy (%)	93.17	95.17	94.03	93.62	92.30	90.68
Average accuracy (%)	89.60	91.55	88.77	89.88	92.92	92.44
Training Time (s)	83.44	74.17	34.14	26.12	4337.41	4104.18
Test Time (s)	2.47	2.32	0.78	0.88	78.61	76.92

Table 19
The results of SVM, NN, 2D-CNN, HybridSN, and FUSENet classifiers with three different dimension reductions of IDA (10, 15, 30) for the four datasets: (1) IPs, (2) KS_CENTER, (3) Dioni, (4) Loukia.

Com.	No.	SVM			NN			2D-CNN			HybridSN		FUSENet		
		10	15	30	10	15	30	10	15	30	15	30	10	15	30
(1)	KA	76.33	76.87	80.61	49.73	53.87	64.43	98.87	99.15	99.69	98.37	99.12	83.88	77.70	75.05
	OA	79.29	79.76	83.05	57.31	60.68	69.15	99.01	99.26	99.73	98.57	99.23	85.94	80.38	78.21
	AA	76.26	75.33	77.04	38.27	41.71	55.09	97.13	99.28	99.45	98.84	98.69	65.68	63.54	63.78
(2)	KA	59.82	59.87	60.30	38.33	40.21	50.02	96.55	99.52	99.15	98.80	99.17	80.52	96.12	8495
	OA	64.16	64.09	64.36	46.31	47.72	55.90	96.90	99.57	99.23	98.92	99.26	82.59	96.52	86.46
	AA	52.42	53.05	56.60	29.31	31.28	41.27	95.26	99.47	98.77	98.42	98.81	81.37	96.83	82.72
(3)	KA	87.97	92.09	93.20	76.46	80.85	87.70	98.78	99.44	99.49	99.24	99.51	98.30	98.20	98.83
	OA	90.30	93.63	94.53	81.20	84.67	90.11	99.01	99.54	99.59	99.39	99.61	98.62	98.54	99.06
	AA	84.63	90.35	90.59	61.12	70.72	85.50	98.10	98.62	99.05	98.19	98.88	97.07	97.94	98.57
(**4**)	KA	69.31	73.71	78.91	58.24	63.35	72.47	92.30	92.71	94.34	92.42	94.06	84.13	88.93	88.63
	OA	74.18	77.88	82.32	65.74	69.72	77.00	93.52	93.86	95.23	93.62	95.00	86.77	90.68	90.47
	AA	69.90	74.20	76.35	46.74	51.17	68.81	91.91	91.40	92.32	89.88	91.96	87.52	92.44	93.18

The following observations resulted from our previous experiments and comparisons: 1) The outcomes depend on more than only the preprocessing and dimension reduction of input data. They rely on the trained model's capacity to extract features and the correct classifications for the preprocessing output. 2) Preprocessing approaches not only play an essential role in boosting the performance time of classifier models and their accuracy but also significantly lower the input data complexity. 3) The results indicate that reducing the outliers, normalizing the distribution, and reducing data complicity are important in the preprocessing stage to enhance the classification stage.

4. Conclusion

This study introduced improving distribution analysis (IDA) as an unsupervised feature reduction method. IDA simplified the data complexity of hyperspectral images and the computational complexity of classification models. IDA consists of three steps that allow it to select the most informative characteristics and deliver exceptional performance. The initial step is to reduce the gap between related data, which reduces the number of outliers and noise. The second step is refining the data and increasing the variance, which improves classification and localization. The final step is to make the distribution more normal. In other words, it reduces differences within a class and increases differences between classes to improve correlation and variance. With the Indian, KS Center, Dioni, and Loukia datasets, the performance of IDA was compared with various popular and state-of-the-art feature extraction approaches. Moreover, IDA's output was also evaluated by several classification methods, including spatial, spectral, and spectral-spatial feature extraction methodologies. The performances and comparisons demonstrated that IDA successfully extracted the features, reduced the dimensionality, and corrected the values' location. It was the most effective solution to the HSI challenges, yielding extremely high accuracy in all classification methods, in some cases exceeding 36%, and reducing computing time.

The IDA was compatible with spatial feature extraction models such as 2DCNN and performed well. Therefore, we must enhance the FRM to be compatible with spectral feature extraction models. In addition, IDA excelled at handling nonlinear datasets such as IPs and KSC, and future work will concentrate on accelerating and improving linear datasets extraction. We observed that, in certain instances, the test time for models using IDA was longer than for models using other DRMs; therefore, we will also focus on reducing the test time.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62150410434).

References

[1] W. Ma, H. Ma, H. Zhu, Y. Li, L. Li, L. Jiao, B. Hou, Hyperspectral image classification based on spatial and spectral kernels generation network, Inf. Sci. (Ny). 578 (2021) 435–456, doi:10.1016/j.ins.2021.07.043.
[2] M. Imani, H. Ghassemian, An overview on spectral and spatial information fusion for hyperspectral image classification: current trends and challenges, Inf. Fusion. 59 (2020) 59–83, doi:10.1016/j.inffus.2020.01.007.
[3] F. Anowar, S. Sadaoui, B. Selim, Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE), Comput. Sci. Rev. 40 (2021) 100378, doi:10.1016/j.cosrev.2021.100378.
[4] G. Chao, Y. Luo, W. Ding, Recent advances in supervised dimension reduction: a survey, Mach. Learn. Knowl. Extr. 1 (2019) 341–358, doi:10.3390/make1010020.
[5] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing 300 (2018) 70–79, doi:10.1016/j.neucom.2017.11.077.
[6] M. Imani, H. Ghassemian, Morphology-based structure-preserving projection for spectral–spatial feature extraction and classification of hyperspectral data, IET Image Process 13 (2019) 270–279, doi:10.1049/iet-ipr.2017.1431.
[7] M. Imani, H. Ghassemian, Feature space discriminant analysis for hyperspectral data feature reduction, ISPRS J. Photogramm. Remote Sens. 102 (2015) 1–13, doi:10.1016/j.isprsjprs.2014.12.024.
[8] V. Nasteski, An overview of the supervised machine learning methods, HORIZONS.B 4 (2017) 51–62, doi:10.20544/HORIZONS.B.04.1.17.P05.
[9] C.O.S. Sorzano, J. Vargas, A.P. Montano, A survey of dimensionality reduction techniques, ArXiv E-Prints. (2014) arXiv:1403.2877. https://ui.adsabs.harvard.edu/abs/2014arXiv1403.2877S.
[10] B. Ghojogh, M.N. Samad, S.A. Mashhadi, T. Kapoor, W. Ali, F. Karray, M. Crowley, Feature selection and feature extraction in pattern analysis: a literature review, ArXiv E-Prints. (2019) arXiv:1905.02845. https://ui.adsabs.harvard.edu/abs/2019arXiv190502845G.
[11] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (2000) 411–430, doi:10.1016/S0893-6080(00)00026-5.
[12] A. Tharwat, Independent component analysis: an introduction, Appl. Comput. Informatic. 17 (2021) 222–249, doi:10.1016/j.aci.2018.08.006.
[13] A.G. Akritas, G.I. Malaschonok, Applications of singular-value decomposition (SVD), Math. Comput. Simul. 67 (2004) 15–31, doi:10.1016/j.matcom.2004.05.005.
[14] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, MIT press Cambridge, 2016 https://amzn.to/2B3MsuU.
[15] L. McInnes, J. Healy, J. Melville, Umap: uniform manifold approximation and projection for dimension reduction, ArXiv Prepr. ArXiv1802.03426. (2018). https://arxiv.org/abs/1802.03426.

[16] Y. Yuan, M. Jin, Multi-type spectral spatial feature for hyperspectral image classification, Neurocomputing (2021), doi:10.1016/j.neucom.2021.12.055.
[17] O. Okwuashi, C.E. Ndehedehe, Deep support vector machine for hyperspectral image classification, Pattern Recognit 103 (2020) 107298, doi:10.1016/j.patcog.2020.107298.
[18] R. Chen, G. Li, Spectral-spatial feature fusion via dual-stream deep architecture for hyperspectral image classification, Infrared Phys. Technol. 119 (2021) 103935, doi:10.1016/j.infrared.2021.103935.
[19] A. Sellami, S. Tabbone, Deep neural networks-based relevant latent representation learning for hyperspectral image classification, Pattern Recognit. 121 (2022) 108224, doi:10.1016/j.patcog.2021.108224.
[20] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, J. Ma, Spectral-Spatial Attention Networks for Hyperspectral Image Classification, Remote Sens. 11 (2019) 963. https://doi.org/10.3390/rs11080963.
[21] C. Pu, H. Huang, L. Yang, An attention-driven convolutional neural network-based multi-level spectral–spatial feature learning for hyperspectral image classification, Expert Syst. Appl. 185 (2021) 115663, doi:10.1016/j.eswa.2021.115663.
[22] X. Wang, Y. Feng, R. Song, Z. Mu, C. Song, Multi-attentive hierarchical dense fusion net for fusion classification of hyperspectral and LiDAR data, Inf. Fusion. 82 (2022) 1–18, doi:10.1016/j.inffus.2021.12.008.
[23] S. Wan, S. Pan, S. Zhong, J. Yang, J. Yang, Y. Zhan, C. Gong, Multi-level graph learning network for hyperspectral image classification, Pattern Recognit. 129 (2022) 108705, doi:10.1016/j.patcog.2022.108705.
[24] L. Yang, F. Zhang, P.S.-P. Wang, X. Li, Z. Meng, Multi-scale spatial-spectral fusion based on multi-input fusion calculation and coordinate attention for hyperspectral image classification, Pattern Recognit. 122 (2022) 108348, doi:10.1016/j.patcog.2021.108348.
[25] K. Karantzalos, C. Karakizi, Z. Kandylakis, G. Antoniou, HyRANK hyperspectral satellite dataset I, (2018). doi:10.5281/zenodo.1222202.
[26] L.E. Christovam, G.G. Pessoa, M.H. Shimabukuro, M. Galo, Land use and land cover classification using hyperspectral imagery: evaluating the performance of spectral angle mapper, support vector machine and random forest, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. (2019).
[27] R. Hang, Z. Li, Q. Liu, P. Ghamisi, S.S. Bhattacharyya, Hyperspectral image classification with attention-Aided CNNs, IEEE Trans. Geosci. Remote Sens. 59 (2021) 2281–2293, doi:10.1109/TGRS.2020.3007921.
[28] M.E. Paoletti, J.M. Haut, J. Plaza, A. Plaza, Deep learning classifiers for hyperspectral imaging: a review, ISPRS J. Photogramm. Remote Sens. 158 (2019) 279–317, doi:10.1016/j.isprsjprs.2019.09.006.
[29] D. Li, Q. Wang, F. Kong, Adaptive kernel sparse representation based on multiple feature learning for hyperspectral image classification, Neurocomputing 400 (2020) 97–112, doi:10.1016/j.neucom.2020.03.022.
[30] J. Fang, X. Cao, Multidimensional relation learning for hyperspectral image classification, Neurocomputing 410 (2020) 211–219, doi:10.1016/j.neucom.2020.05.034.
[31] A. Mohan, M. Venkatesan, HybridCNN based hyperspectral image classification using multiscale spatiospectral features, Infrared Phys. Technol. 108 (2020) 103326, doi:10.1016/j.infrared.2020.103326.
[32] S.K. Roy, G. Krishna, S.R. Dubey, B.B. Chaudhuri, HybridSN: exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification, IEEE Geosci. Remote Sens. Lett. 17 (2020) 277–281, doi:10.1109/LGRS.2019.2918719.
[33] S. Wang, M.E. Celebi, Y.-D. Zhang, X. Yu, S. Lu, X. Yao, Q. Zhou, M.-G. Miguel, Y. Tian, J.M. Gorriz, I. Tyukin, Advances in data preprocessing for biomedical data fusion: an overview of the methods, challenges, and prospects, Inf. Fusion. 76 (2021) 376–421, doi:10.1016/j.inffus.2021.07.001.
[34] Z. Ge, G. Cao, X. Li, P. Fu, Hyperspectral image classification method based on 2D–3D CNN and multibranch feature fusion, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 5776–5788, doi:10.1109/JSTARS.2020.3024841.
[35] S. Roy, R. Mondal, M.E. Paoletti, J.M. Haut, A. Plaza, Morphological convolutional neural networks for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14 (2021) 8689–8702, doi:10.1109/JSTARS.2021.3088228.
[36] H. Guo, J. Liu, J. Yang, Z. Xiao, Z. Wu, Deep collaborative attention network for hyperspectral image classification by combining 2-D CNN and 3-D CNN, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 4789–4802, doi:10.1109/JSTARS.2020.3016739.
[37] M. Khodadadzadeh, X. Ding, P. Chaurasia, D. Coyle, A hybrid capsule network for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14 (2021) 11824–11839, doi:10.1109/JSTARS.2021.3126427.
[38] C. Yu, R. Han, M. Song, C. Liu, C.-.I. Chang, A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 2485–2501, doi:10.1109/JSTARS.2020.2983224.
[39] N. Wambugu, Y. Chen, Z. Xiao, K. Tan, M. Wei, X. Liu, J. Li, Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: a review, Int. J. Appl. Earth Obs. Geoinf. 105 (2021) 102603, doi:10.1016/j.jag.2021.102603.
[40] S.K. Roy, S.R. Dubey, S. Chatterjee, B. Baran Chaudhuri, FuSENet: fused squeeze-and-excitation network for spectral-spatial hyperspectral image classification, IET Image Process 14 (2020) 1653–1661, doi:10.1049/iet-ipr.2019.1462.

Dalal AL-Alimi is currently pursuing her Ph.D. from the University of Geosciences. She received her M.S. degree from the University of Geosciences, Wuhan, China in 2020. Her research interests include remote sensing images, image processing, object detection, image classification, hyperspectral image, deep learning, machine learning, and time series forecasting.

Mohammed A. A. Al-qaness received the B.S., M.S., and Ph.D. degrees from Wuhan University of Technology, in 2010, 2014, and 2017, respectively, all in information and communication engineering. He is an Assistant Professor with the School of Computer Science, Wuhan University, Wuhan, China. He was also a Postdoctoral Follower with the State Key Laboratory for Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. He is currently working at the College of Physics and Electronic Information Engineering, Zhejiang Normal University. His-current research interests include optimization algorithms, wireless sensing, mobile computing, machine learning, signal and image processing, and natural language processing.

Zhihua Cai received the Ph.D. degree in geodetection and information technology from the China University of Geosciences, Wuhan, China, in 2003.,Since 2002, he has been a Professor with the School of Computer, China University of Geosciences. His-research interests include machine learning, evolutionary computation, artificial intelligence, and the processing of remote sensing images.

Eman Ahmed Alawamy is currently pursuing her Ph.D. from Central South University, Hunan, China. She received her M.S. degree from Sana'a University, Sana'a, Yemen in 2013. Her research interests include Markov Chain, qeueing theory, Hamilton Monte Carlo, Bayesian inference, and stochastic model.