Feature Extraction for Hyperspectral Image Classification

M. P. Uddin[1], M. A. Mamun[2] and M. A. Hossain[3]
Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology (RUET), Rajshahi, Bangladesh
[1]palash_cse@hstu.ac.bd, [2]a.mamun@ruet.ac.bd, and [3]ali.hossain@ruet.ac.bd

***Abstract*—Remote sensing hyperspectral image (HSI) contains important information of ground surface as a set of hundreds of narrow and contiguous spectral bands. For effective classification of hyperspectral images, feature reduction techniques through feature extraction and feature selection approaches are applied to improve the classification performance. Principal Component Analysis (PCA) is the widely used feature extraction method for dimensionality reduction. In this paper, PCA and its linear variants such as segmented-PCA (SPCA) and folded-PCA (FPCA) together with nonlinear variants kernel-PCA (KPCA) and Kernel Entropy Component Analysis (KECA) have been studied to effectively extract the features for classification task. The feature selection over the new transformed features was carried out using cumulative-variance accumulation based approach except for KECA that employs Renyi entropy based feature selection. The studied methods are compared using real hyperspectral image. The experimental result shows that the classification accuracy of KPCA (95.9245%) and KECA (95.6262%) outperforms FPCA (95.1292%). However, the FPCA provides the less space complexity.**

Keywords—Feature extraction, feature selection, hyperspectral image classification, PCA, SPCA, FPCA, KPCA, KECA

I. INTRODUCTION

Remote sensing hyperspectral images (HSIs) are captured at large number of narrow and contiguous spectral wavelength bands, which usually range from 0.4μm to 2.5μm covering from visible light to (near) infrared region of electromagnetic spectrum. This facilitates HSIs to use in a wide range of applications including mining, agriculture, geology, and military surveillance. Furthermore, HSI does not only assist in analysis of food quality, pharmaceutical, security and skin but also verifies counterfeit goods and documents [1], [2]. With the spectral resolution in *nm*, HSI provides high discrimination ability in data analysis [2] for various humanitarian facilities such as discrimination among vegetation categories, identification the presence or absence of specific elements in a soil or rock, eutrophication of lakes and so on [3], [4]. Generally, HSI data forms a hypercube that contains 2D spatial information and third dimension offers the spectral information. Thus, the total data contained is $X \times Y \times F$, where F is the number of image bands in the image, X denotes the number of rows and Y denotes the number of columns of each image band. Since the images are usually captured by aircraft or satellite sensors some preprocessing such as atmospheric correction, radiometric correction or geometric correction should be performed [1], [3].

On the other hand, most importantly, although HSIs have hundreds of bands, all the bands may not contain same proportion of information and few bands may contain less discriminatory information since the neighboring bands are highly correlated [5]. Therefore, dimensionality reduction techniques can be applied to select the relevant bands and to extract the intrinsic features through transforming the features with a view to fixing the curse of dimensionality problem or Hughes phenomenon [6]. The dimensionality should be reduced because if the ratio between the number of spectral bands and the number of training sample pixels is not balanced, then the classification accuracy is reduced. Thus, to mitigate this issue feature extraction and feature selection are significantly required [2], [6].

Both supervised and unsupervised feature extraction methods include linear and nonlinear transformation for extracting the appropriate and intrinsic features. Supervised methods require a priori knowledge for extracting the effective features. For instance, Discriminant Analysis Feature Extraction (DAFE) is intended to improve class separability and Decision Boundary Feature Extraction (DBFE) extracts discriminant features to form the decision boundary between classes. These methods could be computationally rigorous and the performance is highly dependent on the training pixels. Alternatively, unsupervised methods are used to extract effective features without using any a priori information [7]. Conversely, the feature selection keeps the original data and selects the meaningful bands based on some searching criteria [8], [9]. However, these search-based methods are sometimes problematic because of combinatorial explosion from high computational cost and local minima problem [10]. Therefore, information-based methods which have the ability to measure both linear and nonlinear relationships between the input image bands and the target classes [11] are often suitable.

FPCA has been theoretically proved as efficient over PCA and SPCA depending on the measure of PSNR, structural similarity index (SSIM), execution time and memory requirement but it may not always offer the best classification accuracy [12]. Conversely, KPCA shows the better performance at a higher computational cost over PCA and ICA (Independent Component Analysis) through a mapping of the input dataset to high dimensional space [13]. In this paper, the unsupervised linear feature extraction methods PCA, SPCA and FPCA together with unsupervised nonlinear feature extraction methods KPCA and KECA are investigated and compared using various performance measure indexes for real hyperspectral image classification.

II. FEATURE EXTRACTION

A. PCA for HSI

PCA is an unsupervised linear feature extraction method that uses orthogonal transformation to identify the correlation among the spectral wavelength bands of HSI for extracting the intrinsic features. To work efficiently, it is based on the assumption that the neighboring bands of HSIs are highly correlated and they often represent the same information about the ground objects.

In its implementation for HSI [14], [15], the HSI hypercube is first converted into a data matrix, denoted as $\mathbf{D}$, of size $F \times S$, where $S=X*Y$. The spectral vector of a pixel, denoted as $\mathbf{x}_n$, in the hypercube or data matrix is defined as $\mathbf{x}_n = [x_{n1} x_{n2} \ldots\ldots x_{nF}]^T$, where, $n \in [1, S]$. Next, the zero-mean image, denoted as $\mathbf{I}$, is produced from $\mathbf{I}_n$ as $\mathbf{I} = [\mathbf{I}_1 \mathbf{I}_2 \ldots.. \mathbf{I}_n]$, where the mean image vector, $\mathbf{M} = \frac{1}{S}\sum_{n=1}^{S} \mathbf{x}_n$ and the mean-adjusted spectral vectors, $\mathbf{I}_n = \mathbf{x}_n - \mathbf{M} = [I_{n1}\, I_{n2} \ldots\ldots I_{nF}]^T$. Then, the covariance matrix, $\mathbf{C} =$

978-1-5386-2175-2/17/$31.00 ©2017 IEEE

$\frac{1}{S}\mathbf{I}\,\mathbf{I}^T$, for the HSI is computed for Eigendecomposition in which eigenvectors ($\mathbf{V_1}$ $\mathbf{V_2}$...$\mathbf{V_F}$), also known as principal components (PCs), and corresponding eigenvalues (E_1E_2... E_F) represent the core of PCA. The eigenvectors determine the directions of the new feature space and the eigenvalues determine their magnitude. In addition, the eigenvalues explain the variance of the data along the new feature axes. If $\mathbf{E}$=*diagonal*(E_1E_2**...** E_F**)** is a diagonal matrix composed by the corresponding eigenvalues E_1, E_2,**...,** E_F of $\mathbf{C}$ and $\mathbf{V}$ denotes the matrix of all respective eigenvectors, then $\mathbf{V}$ is calculated using the following equation:

$$\mathbf{C}=\mathbf{V}\mathbf{E}\mathbf{V}^T \quad (1)$$

Then, k eigenvectors are chosen to form an $F\times k$ dimensional matrix, $\mathbf{w}$, where k is the number of dimensions in the new feature subspace with $k\leq F$ and often $k<<F$. In order to do so, the common approach is to rank the eigenvalues from highest to lowest order to choose the top k eigenvectors. Other approaches include discriminant analysis, divergence analysis e.g., Bhattacharyya distance, mutual information based selection methods etc. Finally, the projection matrix, $\mathbf{Y}$ can be obtained as $\mathbf{Y}=\mathbf{w}^T*\mathbf{I}$. Moreover, the original data matrix with a very few loss, denoted as $\mathbf{d}$, can be obtained as $\mathbf{d}=(\mathbf{w}*\mathbf{Y})+\mathbf{M}$.

B. SPCA for HSI

It has been seen that PCA extracts effective features if the original bands are highly correlated having a big covariance matrix of size $F\times F$ which requires a higher computational cost [16]. However, it is observed that the correlations between neighboring bands of HSI are usually higher than for the bands further apart, with high correlations appearing in blocks. Thus, SPCA is introduced to modify the conventional PCA by avoiding the low correlations between the highly correlated blocks for better performance [16].

In the implementation [16], first the whole zero–mean dataset is divided into L subgroups based either on uniform distribution or on correlation among the bands. Let $\mathbf{D}_i$ denote the subgroup datasets, where $i \in [1, L]$ and n_i be the number of consecutive bands for each subgroup. Then, $\mathbf{D}_1$= [$\mathbf{I}_1'$ $\mathbf{I}_2'$….$\mathbf{I}_n'$], where $\mathbf{I}_j'$ contains first n_1 rows of the respective $\mathbf{I}_j$, $\mathbf{D}_2$= [$\mathbf{I}_1''$ $\mathbf{I}_2''$….$\mathbf{I}_n''$], where $\mathbf{I}_j''$ contains n_2 rows skipping first n_1 rows of the respective $\mathbf{I}_j$ with j, $n \in [1, S]$ and so on. Similarly, $\mathbf{M}_i$ is the mean vector of size $n_i\times 1$ for the respective $\mathbf{D}_i$ that can be found from $\mathbf{M}$. Next, Eigendecomposition is performed on each computed covariance matrix of each $\mathbf{D}_i$. The final projection matrix of the whole dataset is found by merging individual projection matrix for each $\mathbf{D}_i$.

C. FPCA for HSI

Another challenge is to obtain the covariance matrix using PCA and SPCA when the total number of pixels, S is tremendously large, usually over 100K, which often causes software tools such as MATLAB crashed due to memory management problem [2]. Also, PCA treats all bands equally in obtaining the covariance matrix failing to pick up the distinct contributions of the featured F bands and similarly SPCA treats all bands equally in a subgroup [2]. Thus, to reduce the size of covariance matrix and to extract the local structure from the hypercube spectral domain, FPCA is introduced where the correlation between bands as well as band groups are extracted fruitfully [2].

In the implementation [2], every mean-adjusted spectral vector $\mathbf{I}_n$ in the zero–mean dataset is converted into a matrix. Each row of the matrix contains a group of W bands, where the total F bands are folded into H groups or segments. Mathematically, let $\mathbf{A}_n$ be the converted matrix of size $H\times W$ (F=H*W). Then, $\mathbf{A}_n$=[$\mathbf{a}_{n1}$ $\mathbf{a}_{n2}$….$\mathbf{a}_{nH}$]^T, where, $\mathbf{a}_{nh}$=[$I_{n(1+W(h-1))}I_{n(2+W(h-1))}$ …… $I_{n(W+W(h-1))}$] and $h \in [1, H]$. The choice of different values for H and W may affect the results. The covariance matrix, denoted as $\mathbf{C}_n$, for each $\mathbf{A}_n$ is computed as $\mathbf{C}_n = \mathbf{A}_n{}^T\mathbf{A}_n$, where, the size of $\mathbf{A}_n$ is $H\times W$ and that of for $\mathbf{C}_n$ is $W\times W$. Finally, the overall covariance matrix, denoted as $\mathbf{C}_{FPCA}$, for the whole dataset is computed as:

$$\mathbf{C}_{FPCA} = \tfrac{1}{S}\textstyle\sum_{n=1}^{S}\mathbf{C}_n = \tfrac{1}{S}\sum_{n=1}^{S}\mathbf{A}_n{}^T\mathbf{A}_n \quad (2)$$

Thus, the size of the overall covariance matrix is $W\times W$. The projection matrix is then calculated after performing Eigendecomposition on $\mathbf{C}_{FPCA}$.

D. KPCA for HSI

PCA, SPCA and FPCA are not scale invariant and they may not guarantee good class separation in the transformed space since they reduce dimensions only in a linear way. Moreover, if the spectral data contains more complicated structures that cannot be well represented in a linear subspace, they may not be too useful [7], [17]. Fortunately, KPCA permits to generalize conventional PCA for nonlinear dimensionality reduction. Each spectral vector $\mathbf{x_i}$ is projected to a new vector $\emptyset(\mathbf{x}_i)$ in a new higher dimensional feature space, where $\emptyset$ is a nonlinear function. Conventional PCA can be performed in the new feature space, but this can be tremendously costly and inefficient. Thus, kernel methods can be used to simplify the computation.

In the implementation, first the kernel matrix $\mathbf{K}$ is constructed from $\{\mathbf{x}_i\}$ as $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, where, $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function which is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \emptyset(\mathbf{x}_i)^T\emptyset(\mathbf{x}_j)$. Then, the Gram matrix $\widetilde{\mathbf{K}}$ is computed using the following equation.

$$\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_S\mathbf{K} - \mathbf{K}\mathbf{1}_S + \mathbf{1}_S\mathbf{K}\mathbf{1}_S, \quad (3)$$

where, $\mathbf{1}_S$is the $S\times S$ matrix with all elements equal to $1/S$. Next, the equation is used to solve for the vectors $\mathbf{a}_i$ ($\mathbf{K}$ is substituted with $\widetilde{\mathbf{K}}$ if the projected dataset $\{\emptyset(\mathbf{x}_i)\}$ does not have zero mean.).

$$\mathbf{K}\mathbf{a}_k = \mathbf{E}_k N\mathbf{a}_k, \quad (4)$$

where, $\mathbf{a}_k$ is the S dimensional column vector of a_{ki} i.e., $\mathbf{a}_k$ = [$a_{k1}a_{k2}$ ….. a_{kS}]^T. The kernel principal components $\mathbf{y}_k(\mathbf{x})$ is then computed using the following equation.

$$\mathbf{y}_k(\mathbf{x}) = \emptyset(\mathbf{x})^T\mathbf{v}_k = \textstyle\sum_{i=1}^{S} a_{ki}k(\mathbf{x}, \mathbf{x}_i) \quad (5)$$

The power of kernel methods is that $\emptyset(\mathbf{x}_i)$ should not be calculated explicitly. The kernel matrix can directly be constructed from the training dataset $\{\mathbf{x}_i\}$. The polynomial kernel, $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^d$ or $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + c)^d$, where c > 0 and the Gaussian kernel or Radial Basis Function (RBF), $k(\mathbf{x}, \mathbf{y}) = exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2})$ with parameter σ that can directly influence on the performance are the commonly used kernels. Ideally, σ should be smaller than the interclass distances and larger than the inner-class distances if we want to separate classes in the new feature space. Since how many classes are there in the dataset is unknown while applying KPCA, thus it is uneasy to estimate the distances. Alternatively, the value of σ can be set to a small value to capture only the neighborhood information of each data point. For the same, let the distance from each data point $\mathbf{x_i}$ to its nearest neighbor be d_i^{NN}. Then, one way to find the parameter is $\sigma = a.mean\,(d_i^{\mathrm{NN}})$, where, a is an arbitrary integer. This strategy is used in the experiment. The detail discussion on parameter selection and pre-image reconstruction is in [17].

E. KECA for HSI

KECA, a successor of KPCA, reveals structure based on Renyi quadratic entropy of the input space dataset thus not on the eigenvalues and eigenvectors of the kernel matrix directly [18], [19]. This method performs data transformation and dimensionality reduction by projecting onto those KPCA axes which give contribution to the entropy estimate. Usually, the axes will not essentially correspond to the highest eigenvalues and eigenvectors of the kernel matrix as in the case for KPCA dimensionality reduction. Thus, KECA may produce strikingly different transformed data than KPCA. In the implementation [18], [19], as like KPCA, the kernel matrix $\mathbf{K}$ from $\{\mathbf{x}_i\}$ is

constructed. Then, $V(p) = \int p^2(\mathbf{x})d\mathbf{x}$ is in concentration since the logarithm of the following original Renyi quadratic entropy estimation used in KECA is a monotonic function.

$$H(p) = -\log \int p^2(\mathbf{x})d\mathbf{x}, \quad (6)$$

where, $p(\mathbf{x})$ is the probability density function (*pdf*) generating the dataset, $\mathbf{D}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]$ of dimensionality d. Then, the Renyi entropy estimator, $\hat{V}(p)$ in terms of the eigenvalues and eigenvectors of the kernel matrix is calculated that may be eigendecomposed as $\mathbf{K}=\mathbf{EDE}^T$, where, $\mathbf{D}$ is a diagonal matrix that stores the eigenvalues $\lambda_1,\ldots..,\lambda_N$ and $\mathbf{E}$ is the matrix with the corresponding eigenvectors $\mathbf{e}_1,\ldots,\mathbf{e}_N$ as columns using the following equation:

$$\hat{V}(p) = \frac{1}{N^2}\sum_{i=1}^{N}(\sqrt{\lambda_i}\mathbf{e}_i{}^T\mathbf{1})^2 \quad (7)$$

Using KPCA, the projection of defined Φ onto the i^{th} principal axis $\mathbf{u}_i$ in the kernel feature space is as $P_{\mathbf{u}_i}\Phi = \sqrt{\lambda_i}\mathbf{e}_i{}^T$. Since KECA is defined as a k-dimensional data transformation obtained by projecting Φ onto a subspace U_k spanned by those k KPCA axes contributing most to the Renyi entropy estimate of the data. Hence, a subset of KPCA axes but not necessarily those corresponding to the top k eigenvalues constitutes the U_k. The transformed data, Φ_{eca} for KECA is calculated as follows:

$$\Phi_{eca} = P_{U_k}\boldsymbol{\varphi} = \sqrt{\mathbf{D}_k}\mathbf{E}_k{}^T \quad (8)$$

III. Experimental Result

A. Dataset Decription

The Indian Pine HSI acquired by Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pine test site in North-western Indiana with 220 spectral reflectance bands is used in the experiment. Each band image is of 145×145 size with 20 meters geometric resolution/ground sampling distance (GSD). The wavelength range is 400-2500 *nm* (spectral resolution: 10 *nm*) having 16 ground classes [20].

B. Performance Measure

Cumulative variance, peak signal to noise ratio (PSNR) and memory requirement have been calculated as a performance measure. However, to evaluate the methods for HSI, overall classification accuracy has been considered. 2127 pixels of 14 different classes from the dataset have been used.

Twelve (12) principal components are selected to accumulate approximately 100% cumulative variance for PCA, SPCA and FPCA to discuss variance information and PSNR. However, the cumulative variance using 12 features of KPCA is 63.34% since this method requires comparatively more, often all features to accumulate approximately 100% cumulative variance. Moreover, Renyi quadratic entropy measure is used for feature selection from KECA transformation.

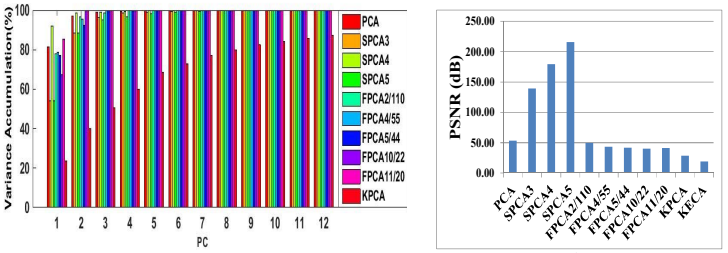
TABLE I. Segmentation of the Complete Set of Bands

Sub-group	Parameter	Number of segments		
		3 (SPCA3)	*4 (SPCA4)*	*5 (SPCA5)*
1	Bands	1-36	1-36	1-36
	Average correlation in diagonal	0.9365	0.9365	0.9365
	Selected PCs	1-6	1-5	1-5
2	Bands	37-102	37-79	37-79
	Average correlation in diagonal	0.6263	0.9421	0.9421
	Selected PCs	1-4	1-3	1-3
3	Bands	103-220	80-102	80-102
	Average correlation in diagonal	0.7024	0.943	0.943
	Selected PCs	1-2	1-2	1-2
4	Bands	N/A	103-220	103-162
	Average correlation in diagonal		0.7024	0.5196
	Selected PCs		1-2	1
5	Bands	N/A	N/A	163-220
	Average correlation in diagonal			0.941
	Selected PCs			1

Based on the correlation matrix of the dataset SPCA is applied in 3 different ways with 3, 4 and 5 segments respectively of varying band-subgroups denoted as SPCA3, SPCA4 and SPCA5 as shown in the Table I. FPCA has been applied on the dataset in five (5) different ways with five different folding options which is shown in Table II. KPCA and KECA are applied with RBF kernel. The PCs versus cumulative variances (%) and PSNR are shown in Fig. 1, where it is seen that PCA converges fast to approximately 100% variance and SPCA produces better summation of PNSR of the segments.

TABLE II. Folding Options for FPCA

Option	H	W
1 (FPCA2/110)	2	110
2 (FPCA4/55)	4	55
3 (FPCA5/44)	5	44
4 (FPCA10/22)	10	22
5 (FPCA11/20)	11	20



(a) (b)

Fig. 1. (a) PCs versus cumulative variance (%), (b) PSNR

However, memory requirement for the methods over different stages is compared in the Table III, where it is clear that KPCA and KECA require more memory for both data and covariance matrices whereas FPCA requires the least in both stages.

TABLE III. Memory Requirement

Method	Stage	
	Data Matrix	*Covariance Matrix*
PCA	$S\times HW$	$HW\times HW$
SPCA	$S\times W$	$W\times W$
FPCA	$H\times W$	$W\times W$
KPCA	$S\times HW$	$S\times S$
KECA	$S\times HW$	$S\times S$

C. Classification Accuracy

To evaluate the performance of the feature extraction methods, classification accuracy has been calculated when support vector machine (SVM) with RBF kernel is applied. MATLAB LibSVM package is used [21]. The 10-fold cross validation scheme has been used to select the *C* and *gamma* parameters for the efficient training and testing. The training set contains 1121 pixels and the testing set contains 1006 pixels of 14 different classes as illustrated in Table IV.

TABLE IV. Detail of the Training and Testing Samples

Class name	Training samples	Testing samples
Hay-windrowed	165	135
Soybean-notill	109	85
Woods	279	248
Wheat	42	63
Grass-trees	96	80
Soybean-min	130	165
Corn-min	108	72
Stone-Steel-Towers	48	40
Buildings-Grass-Trees-Drives	48	44
Grass-pasture	15	14
Corn-notill	25	20
Soybean-clean	15	15
Corn	21	10
Alfalfa	20	15

Feature extraction methods for classification suffer from the major problem of searching the optimal number of PCs. In this experiment, the first PC calculated using the largest

eigenvalue/Renyi entropy is used as the input of SVM to perform the classification. Then, the PCs are increased one by one corresponding to the largest eigenvalue/Renyi entropy at each step [13] and the overall classification accuracy (OA) is noticed. The graph in Fig. 2 shows the OA using the feature extraction methods with every features for the dataset.

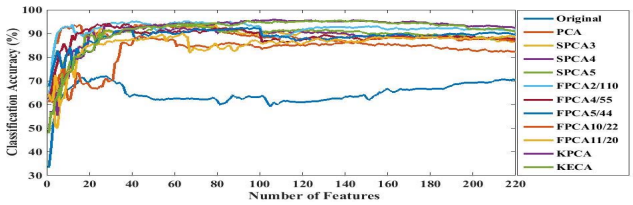


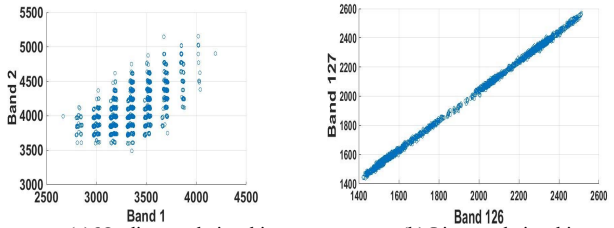
Fig. 2. Classification accuracy versus features

The optimal number of PCs that produces the best OA using various feature extraction methods is illustrated in Table V.

TABLE V. OVERALL CLASSIFICATION ACCURACY

Method	Optimal number of image bands	Overall Accuracy (%)
PCA [14]	46	93.7376
SPCA3 [16]	79	93.837
SPCA4 [16]	50	93.7376
SPCA5 [16]	51	93.9364
FPCA2/110 [2]	40	95.1292
FPCA4/55 [2]	29	94.4334
FPCA5/44 [2]	38	92.3459
FPCA10/22 [2]	43	88.7674
FPCA11/20 [2]	64	90.2584
KPCA [17]	107	95.9245
KECA [18]	147	95.6262

From the above table, it can be seen that any feature extraction method produces better accuracy than using the original dataset, which is 70.2783%. Moreover, KPCA with RBF kernel in which σ=0.019 produces the best OA. Since several bands of the dataset are nonlinear to each other as shown in Fig. 3 of scatter plots, KPCA as a nonlinear method fixes the nonlinearity for obtaining better classification accuracy.



(a) Nonlinear relationship (b) Linear relationship

Fig. 3. (a) Scatter plot of bands 1 and 2, (b) Scatter plot of bands 126 and 127

On the other hand, KECA produces strikingly different transformed data than KPCA and with RBF kernel having σ=0.011 tries to give the treatments for the nonlinearity to produce better OA but not like KPCA. Moreover, although FPCA2/110 is a linear technique, its OA with comparatively fewer features is very near to that of KECA and KPCA. The reason is that both global and local structure of the dataset is finely extracted from the bands of each of the 2 folds with 110 bands for each. At last, it is observed that SPCA and FPCA attempt to perform better over PCA as they deal with the extraction of local characteristics in different ways.

IV. CONCLUSION

To extract the intrinsic features from the HSI, PCA, SPCA, FPCA, KPCA and KECA are discussed in terms of their working steps, variance accumulation, PSNR etc. The feature extraction methods are compared in terms of memory requirement and overall classification accuracy with the mostly used feature selection approach i.e., variance based selection for PCA, SPCA, FPCA and KPCA and Renyi quadratic entropy based selection for KECA. Nonlinear methods KPCA and KECA produce better classification accuracy for the dataset at a higher cost of memory and computational time. On the other hand, FPCA is the memory saving feature extraction methods that produces very close OA to KECA and KPCA for the dataset with H=2 and W=110. However, if the settings such as segmentation criteria for SPCA and FPCA, the kernel trick for KPCA and KECA etc. are changed, then the classification accuracy may differ. It can further be concluded that FPCA can also extract the local characteristics in the case of several nonlinear relationships among the bands and the nonlinear methods fix the nonlinearity for better classification.

REFERENCES

[1] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, 4th edition, Berlin, Germany: Springer-Verlag, 2006.

[2] J. Zabalza, J. Ren, M. Yang, Y. Zhang, J. Wang, S. Marshall, and J. Han, "Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," *ELSEVIER ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp.112–122, 2014.

[3] B. K. Mohan and A. Porwal, "Hyperspectral image processing and analysis," *Current Science*, vol. 108, no.5, pp. 833-841, 2015.

[4] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gürbüz and U. Sakarya, "A Short Survey of Hyperspectral Remote Sensing Applications in Agriculture," *6th IEEE International Conference on Recent Advances in Space Technologies (RAST)*, 2013.

[5] B. Guo, S. R. Gunn, R. I. Damperand and J. D. B. Nelson, "Band Selection for Hyper spectral Image Classification Using Mutual Information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, 2006.

[6] G. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Transactions on Information Theory*, vol. IT-14, no.1, pp. 55–63, 1968.

[7] M. Fauvel, J. Chanussot and J. A. Benediktsson, "Kernel Principal Component Analysis for Feature Reduction in Hyperspectral Image Analysis," *IEEE 7th Nordic Signal Processing Symposium (NORSIG),* pp. 238-241, Iceland, 2006.

[8] X. Jia, B. Kua, and M. M. Crawford, "Feature Mining for Hyperspectral Image Classification," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676-679, 2013.

[9] C. Chang, D. Qian, T. L. Sun, and M. Althouse, "A Joint Band Prioritization and Band-decorrelation Approach to Band Selection for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2631–2641, 1999.

[10] F. Melgani and L. Bruzzone, "Classification of Hyperspectral Remote Sensing Images with Support Vector Machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[11] H. Peng, F. Long and C. Ding, "Feature Selection Based on Mutual Information Criteria of Max-dependency, Max-relecvance, and Min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp.1226–1238, 2005.

[12] P. Deepa and K. Thilagavathi, "Data Reduction Techniques of Hyperspectral Images: A Comparative Study," *IEEE 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015.

[13] L.J. Cao, K.S. Chua, W.K. Chong, H.P. Lee and Q.M. Gu, "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine," *ELSEVIER Neurocomputing*, vol. 55, issues. 1–2, pp. 321-336, 2003.

[14] C. Rodarmel and J. Shan, "Principal Component analysis for hyper-spectral image classification," *ACM Surveying and Land Information Science*, vol. 62, no. 2, pp. 115-122, 2002.

[15] A. Shabna, and R. Ganesan, "HSEG and PCA for Hyper-spectral Image Classification," *IEEE International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* 2014.

[16] X. Jia, and J. A. Richards, "Segmented Principal Components Transformation for Efficient Hyperspectral Remote-Sensing Image Display and Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 538-542, 1999.

[17] Q. Wang, "Kernel Principal Component Analysis and its Application in Face Recognition and Active Shape Models," *ARXIV1207.3538*, 2012.

[18] R. Jenssen, "Kernel Entropy Component Analysis," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 847-860, 2010.

[19] L. Gómez-Chova, R. Jenssen and G. Camps-Valls, "Kernel Entropy Component Analysis for Remote Sensing Image Clustering," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, No. 2, pp. 312-316, 2012.

[20] M. F. Baumgartner, L. Labial and D. A Landgrebe, "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3," *Purdue University Research Repository*, 2015.

[21] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, No. 3, pp. 27:1–27:27, 2011.