Differential Evolution Extreme Learning Machine for the Classification of Hyperspectral Images

Yakoub Bazi, *Senior Member, IEEE*, Naif Alajlan, *Senior Member, IEEE*, Farid Melgani, *Senior Member, IEEE*, Haikel AlHichri, *Member, IEEE*, Salim Malek, *Member, IEEE*, and Ronald R. Yager, *Life Member, IEEE*

***Abstract*—Recently, a new machine learning approach that is termed as the extreme learning machine (ELM) has been introduced in the literature. This approach is characterized by a unified formulation for regression, binary, and multiclass classification problems, and the related solution is given in an analytical compact form. In this letter, we propose an efficient classification method for hyperspectral images based on this machine learning approach. To address the model selection issue that is associated with the ELM, we develop an automatic-solution-based differential evolution (DE). This simple yet powerful evolutionary optimization algorithm uses cross-validation accuracy as a performance indicator for determining the optimal ELM parameters. Experimental results obtained from four benchmark hyperspectral data sets confirm the attractive properties of the proposed DE-ELM method in terms of classification accuracy and computation time.**

***Index Terms*—Differential evolution (DE), extreme learning machine (ELM), feature extraction, hyperspectral images.**

I. Introduction

In RECENT years, the issue of the supervised classification of hyperspectral images has attracted much research from the remote sensing community. Compared with previous developments [1]–[3], the current objective that is set by researchers is to further boost the classification accuracy [4]–[11]. From these new works, one can discern that an important direction of the research is related to the choice of classification approach. In addition to the state-of-the-art support vector machines (SVMs), other learning methods that are made available by the machine learning community have been also investigated.

Recently, a new machine learning approach that is termed as the extreme learning machine (ELM) has been introduced in the literature [12], [15]. The ELM aims at minimizing the training error and the norm of the output weights. Compared with the existing learning methods, the ELM is characterized by a unified formulation for binary, multiclass, and regression problems. The solution for these problems is also given in a unified analytical compact form. In the ELM, feature mapping could be done either in a known space that is similar to neural networks or in an infinite space that is similar to kernel methods. For binary classification, a single output node is used, and the decision function is based on the sign function, as used for the SVM. In multiclass classification, the ELM uses a configuration of multioutput nodes, where the number of nodes is equal to the number of classes. The assignment of a test sample to a particular class is done by identifying the index of the output node with the highest decision value. It is worth mentioning that the ELM was first proposed for the single hidden-layer feedforward neural networks (SLFNs). Then, it was extended to the generalized SLFNs, where the hidden layer needs not be neuron-like [12], [13].

In this letter, we propose to investigate the capabilities of the ELM for the classification of hyperspectral images, since it is well known that the integration of spatial contextual information in the learning process will lead to enhanced classification results. We consider, in this letter, the solution that is based on morphological profiles (MPs) [6]. However, any appropriate feature extraction method could be considered as well. As hyperspectral images are characterized by high-dimensional spectral features, we first apply feature reduction [(such as the principal component analysis (PCA)] to reduce the dimensionality of the data. Then, in the second step, we apply morphological operations (opening and closing operations with reconstruction) to these features to generate an extra set of MP features. To address the model selection issue that is associated with the ELM, simple grid selection procedures could be used. Nonetheless, in order to achieve better search performance, sophisticated solutions based on evolutionary computation could be considered as well [3], [15], [17]. In particular, in this letter, we adopt a solution that is based on the differential evolution (DE) algorithm [15]. This simple yet powerful evolutionary optimization algorithm uses cross-validation accuracy as a performance indicator for determining the optimal ELM parameters.

II. Proposed Classification Method

A. Classification With ELM

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be the training set that is composed of N training feature vectors $\mathbf{x}_i$ of dimension d, and $y_i \in \{1, \ldots, P\}$ are class labels. P represents the number of classes. Each vector $\mathbf{x}_i$ is composed of two parts: the N_{PCA} features obtained by applying the PCA to the original spectral features and the MP features obtained by applying the opening and

Manuscript received June 3, 2013; revised September 24, 2013; accepted October 10, 2013. Date of publication November 6, 2013; date of current version January 28, 2014. This work was supported by the Distinguished Scientist Fellowship Program of King Saud University.

Y. Bazi, N. Alajlan, H. AlHichri, and S. Malek are with the Advanced Laboratory for Intelligent Systems Research Laboratory, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: ybazi@ksu.edu.sa; najlan@ksu.edu.sa; hhichri@ksu.edu.sa).

F. Melgani is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: melgani@disi.unitn.it).

R. R. Yager is with the Machine Intelligence Institute, Iona College, New Rochelle, NY 10801 USA, and also with King Saud University, Riyadh 11451, Saudi Arabia (e-mail: yager@panix.com).

Digital Object Identifier 10.1109/LGRS.2013.2286078

closing operations with reconstruction to these N_{PCA} features with a structural element of different sizes.

The jth output of a multiclass ELM classifier with P output nodes is given by

$$f_j(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{w}_j, \quad j = 1, \ldots, P \tag{1}$$

where $\mathbf{w}_j \in \boldsymbol{R}^L$ is the vector of the output weights between the hidden layer of L nodes and the jth output node. In the case of binary classification, the ELM is characterized by one output node. $\mathbf{h}(\mathbf{x}) \in \Re^L$ is the (row) output vector of the hidden layer with respect to input $\mathbf{x}$. It maps the input data from the d-dimensional space to the L-dimensional ELM feature space. This feature mapping could be done in a finite space, such as in standard neural network classifiers, or in an infinite space by applying the kernel trick, as shown later.

The $l2$ norm optimization problem that is associated with the ELM is given as follows:

$$\begin{aligned} &\text{Minimize} \quad L_{\mathrm{Primal}} = \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &\text{Subject to} \quad \mathbf{h}(\mathbf{x}_i)\mathbf{w} = \boldsymbol{\eta}_i^{\mathbf{T}} - \boldsymbol{\xi}_i^{\mathbf{T}}, \quad i = 1, \ldots, N \end{aligned} \tag{2}$$

where C is a regularization parameter. $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_P]$ is a matrix of size $P \times L$ formed by staking the vectors of output weights $\mathbf{w}_j, j = 1, \ldots, P$. $\boldsymbol{\eta}_i = [\eta_{i1}, \ldots, \eta_{iP}]^{\mathbf{T}}$ and $\boldsymbol{\xi}_i = [\xi_{i1}, \ldots, \xi_{iP}]^{\mathbf{t}}$ are the target and training error vectors of the P output nodes, respectively, with respect to the training sample $\mathbf{x}_i$. Target vector $\boldsymbol{\eta}$ has all its values set to 0, except the entry that matches the class label y_i, which is set to 1. Based on the Karush–Kuhn–Tucker (KKT) theorem, the determination of the weights training an ELM is equivalent to solving the following dual optimization problem:

$$\begin{aligned} L_{\mathrm{Dual}} =& \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &- \sum_{i=1}^{N}\sum_{j=1}^{p}\alpha_{ij}\left(\mathbf{h}(\mathbf{x_i})\mathbf{w}_j - \eta_{ij} + \xi_{ij}\right). \end{aligned} \tag{3}$$

It can be shown that by taking the KKT optimal conditions, the optimal vector of weights $\mathbf{w}^*$ can be given by the following compact matrix form [11]:

$$\mathbf{w}^* = \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\boldsymbol{\eta} \tag{4}$$

where $\mathbf{H}$ is the hidden-layer output matrix, and it is defined as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix} \tag{5}$$

$\boldsymbol{\eta}$ is a matrix of size $N \times P$ built from target output vector $\boldsymbol{\eta}_i^T$, as in the following:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1^T \\ \vdots \\ \boldsymbol{\eta}_N^T \end{bmatrix} = \begin{bmatrix} \eta_{11} & \cdots & \eta_{1P} \\ \vdots & \ddots & \vdots \\ \eta_{N1} & \cdots & \eta_{NP} \end{bmatrix} \tag{6}$$

and $\mathbf{I}$ is an identity matrix of size $N \times N$. Hence, the output function of the ELM is

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{w}^* = \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\boldsymbol{\eta}. \tag{7}$$

During the prediction phase, test sample $\mathbf{x}_{\boldsymbol{\ell}}$ will be assigned to the index of the output node that has the highest value. In other words, if we let $\mathbf{f}(\mathbf{x}_{\boldsymbol{\ell}}) = [f_1(\mathbf{x}_{\boldsymbol{\ell}}), \ldots, f_P(\mathbf{x}_{\boldsymbol{\ell}})]^T$, then the predicted class label for test sample $\mathbf{x}_{\boldsymbol{\ell}}$ is

$$y_\ell^* = \arg\max_{j \in \{1,\ldots,P\}} f_j(\mathbf{x}_\ell). \tag{8}$$

In the kernel space, the prediction that is associated with test sample $\mathbf{x}_{\boldsymbol{\ell}}$ is given in the following compact form:

$$\mathbf{f}(\mathbf{x}_\ell) = \begin{bmatrix} k\left(\mathbf{x}_\ell, \mathbf{x_1}\right) \\ \vdots \\ k\left(\mathbf{x}_\ell, \mathbf{x}_N\right) \end{bmatrix}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C} + \mathbf{K}\right)^{-1}\boldsymbol{\eta}. \tag{9}$$

The first term of (9) is a vector of length N, and it represents the kernel distances between test point $\mathbf{x}_{\boldsymbol{\ell}}$ and the training samples. In this case, the number of hidden neurons L need not be given as kernel matrix $\mathbf{K} = \mathbf{H}\mathbf{H}^T$, which is only related to the training samples. It is not relevant to the number of output nodes and to the training target values. From (9), one can clearly see that training and predicting with the ELM is done in a simple way.

It is interesting to notice that the prediction equation of the ELM is similar to the prediction equation of the Gaussian process regression (GPR) [16], [17]. However, the differences could be summarized in the following main points: The GPR formulation is done under a Bayesian framework and only deals with regression problems, and $\boldsymbol{\eta}$ is a real target vector that is associated with the input training samples, whereas in the ELM, it represents a binary matrix.

See [14] for a detailed description of the ELM and its comparison with other classifiers.

B. Model Selection With DE (DE-ELM)

DE is a powerful evolutionary algorithm for global numeric optimization [18], [19]. Compared with most of the available evolutionary algorithms, it is much simpler and straightforward to implement. It maintains a population of NP members, where each member (called a target vector) represents a solution in space $\mathcal{F}$. Then, it improves this population generation by generation. In general, DE and its variants are composed of the following sequential steps: mutation, crossover, and selection.

Let us consider $\vec{\mathbf{v}}_i \in \boldsymbol{R}^D$, $i = 1, \ldots, NP$ as a target vector of size D from solution space $\mathcal{F}$, where NP is the population size. Let $\vec{\mathbf{u}}_i$ and $\vec{\mathbf{q}}_i$ also be its mutant and trial vectors, respectively. Mutant vector $\vec{\mathbf{u}}_i$ is generated during the mutation step as follows:

$$\vec{\mathbf{u}}_i = \vec{\mathbf{v}}_{r_1} + F \cdot (\vec{\mathbf{v}}_{r_2} - \vec{\mathbf{v}}_{r_3}) \tag{10}$$

where r_1, r_2, and r_3 are mutually exclusive integers that are randomly chosen from the range [1, NP], and F is a control parameter that is often called the scaling parameter (typically, it lies in the interval [0.4, 1]). In the next step, trial vector $\vec{\mathbf{q}}_i$

is generated by applying a crossover operator to vectors $\vec{\mathbf{v}}_i$ and $\vec{\mathbf{q}}_i$. Two widely common operators that are used in DE implementations are the binomial and exponential operators. Here, we consider the binomial crossover operator. Under this scheme, the trial vector is generated as follows:

$$\vec{q}_{ij} = \begin{cases} v_{ij} & \text{if } \text{rand}_j[0,1] \leq \text{CR} \quad \text{or } j = j_{\text{rand}} \\ u_{ij} & \text{otherwise} \end{cases} \tag{11}$$

where j_{rand} is a randomly chosen integer in the range [1, D], rand is a uniform random number in the range [0, 1], and CR $\in [0, 1]$ is called the crossover control parameter. To keep the population size constant over the generations, the DE algorithm calls for a selection step whether to keep the target or the trial vectors during the next generation. In the case of minimization, the vector with the smallest objective function is kept.

In this letter, we propose to apply the DE to solve the model selection problem of the ELM. To this end, the target vector represents the regularization and kernel parameters. In the case of the radial basis function kernel, this vector is given as follows:

$$\vec{\mathbf{v}}_i = [C\ \gamma]\,. \tag{12}$$

As the objective function, we propose to minimize the widely used cross-validation error measure, as in the following:

$$\mathbf{g}(\vec{\mathbf{v}}_i) = \text{CV}_{\text{err}}. \tag{13}$$

Such error is computed by splitting the training set during the training phase into k-folds and then by training the ELM on k-1 folds and computing the error on the remaining fold. This operation is done for all possible fold combinations, and then, the average error is taken [3]. We provide the main steps of the proposed DE-ELM algorithm in the following. It is interesting to note that, in order to enhance the search ability of the DE, we use a recent variant of it, which includes an additional crossover operation called the orthogonal crossover [19].

Algorithm: DE-ELM

Input: - Training set $\mathcal{D}$
- DE parameters: NP, CR, F, and the maximum number of function evaluations $FES_{\max}$

Output: - Classification result

Step 1) Initialization:

Step 1.1) Set the index of generations Gen $= 0$

Step 1.2) Generate random target vectors $\vec{\mathbf{v}}_{i,\text{Gen}}$, $i=1, \ldots, NP$ from solution space $\mathcal{F}$ to form an initial population of size NP.

Step 1.3) For each vector $\vec{\mathbf{v}}_{i,\text{Gen}}$, run an ELM classifier on the training set and compute the corresponding objective function $\boldsymbol{g}(\vec{\mathbf{v}}_{i,\text{Gen}})$. Set the number of function evaluations $FES = NP$.

Step 2) *for* $i = 1$ to NP, do the following:

Step 2.1) *Mutation step*: Generate the mutant vector $\vec{\mathbf{u}}_{i,\text{Gen}}$ corresponding to target vector $\vec{\mathbf{v}}_{i,\text{Gen}}$ via the DE scheme (10).

Step 3.2) *Crossover step*: Generate trial vector $\vec{\mathbf{q}}_{i,\text{Gen}}$ through the binomial crossover according to (11).

Step 3.3) *Selection step*: Compute the objective function $\boldsymbol{g}(\vec{\mathbf{q}}_{i,\text{Gen}})$ corresponding to trial vector $\vec{\mathbf{q}}_{i,\text{Gen}}$ if $\boldsymbol{g}(\vec{\mathbf{q}}_{i,\text{Gen}}) \leq \boldsymbol{g}(\vec{\mathbf{v}}_{i,\text{Gen}})$, *then* $\vec{\mathbf{v}}_{i,\text{Gen+1}}) = \vec{\mathbf{q}}_{i,\text{Gen}}$ *else* $\vec{\mathbf{v}}_{i,\text{Gen+1}} = \vec{\mathbf{v}}_{i,\text{Gen}}$ *end if*

end for

Step 4) Set $FES = FES + NP$

Step 5) Go to Step 7 if $FESgeqFES_{\max}$, otherwise set Gen $=$ Gen $+ 1$ and return to step 2.

Step 6) Select the optimal vector $\vec{\mathbf{v}}_i^*$ corresponding to the minimum objective function $\boldsymbol{g}(\vec{\mathbf{v}}_i^*)$.

Step 7) Train the DE-ELM using the optimal parameter vector $\vec{\mathbf{v}}_i^*$, and compute the decision function for test sample $\mathbf{x}_\ell$ according to (9).

TABLE I
HYPERSPECTRAL DATA SETS USED IN THE EXPERIMENTS

Dataset	Image size	Spatial Res.	#Bands	(Train, Test)	#Classes
Indiana	145×145	20 m	220	(695, 9671)	16
KSC	512×614	18 m	176	(650, 8052)	13
Wahsington DC	1280×307	2 m	191	(350, 56919)	7
Pavia	610×340	1.3 m	103	(450,42326)	9

III. EXPERIMENTAL RESULTS

A. Data Set Description and Performance Evaluation

To assess the effectiveness of the ELM classification method, low spatial resolution and high spatial resolution data sets are used in the experiments, as shown in Table I. See [2] and [11] for a detailed description of these data sets.

Indiana: Acquired in 1992 over the Indian Pines test site in Northwestern Indiana by the AVIRIS sensor. This data set consists of 16 land cover classes.

Kennedy Space Center (KSC), Florida: Acquired by the NASA AVIRIS instrument in 1996 from an altitude of approximately 20 km. This image is composed of 13 land cover classes representing various land cover types.

Washington, DC: Acquired by the hyperspectral digital imagery collection experiment (HYDICE) from an airborne hyperspectral data flight line over the Washington, DC urban area. It consists of seven land cover classes.

University of Pavia, Pavia, Italy: Acquired by the reflective optics system imaging spectrometer (ROSIS-03) optical sensor in 2002 over the University of Pavia. Nine classes characterize this image.

In the experiments, for all data sets, we consider 50 training samples per class, and the remaining samples are left for the test. For Indiana, we also consider 50 training samples per class, except for the minority classes *alfalfa*, *grass/pasture-mowed*, and *oats*, where we only use 15 training samples per class [4], [11]. For all the data sets, we repeat the experiments ten times with different training and test samples, and then, we present the averaged results in terms of the overall (OA) and average (AA) standard deviation, i.e., σ_{OA} and σ_{AA}, respectively, and the computation time that is related to the model selection and the classifier training with the obtained optimal parameters. In addition to these measures, we also use McNemar's statistical

TABLE II
CLASSIFICATION RESULTS OBTAINED FOR THE LOW SPATIAL RESOLUTION DATA SETS. (a) AVIRIS AND (b) KSC DATA SETS. THE DIFFERENCE AT A LEVEL OF SIGNIFICANCE OF 5% ($Z_{\mathrm{ELM,SVM}} > 1.96$) IS HIGHLIGHTED IN BOLDFACE

(a)

Features	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
Spectral	79.36±1.45 87.90±0.57	18	**80.97±1.12** **87.35±0.99**	296	-0.88
$\mathrm{MP}^{\mathrm{PCA(5)}}$	**93.74±0.67** **96.36±0.54**	14	93.04±0.80 95.52±1.08	112	**3.64**
$\mathrm{MP}^{\mathrm{PCA(10)}}$	**95.25±0.70** **97.59±0.21**	14	94.67±0.75 97.05±0.30	180	**3.52**

(b)

Features	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
Spectral	91.46±0.68 88.42±0.66	15	**92.87±0.36** **89.79±0.75**	108	**-2.69**
$\mathrm{MP}^{\mathrm{PCA(5)}}$	**98.07±0.40** **97.80±0.54**	12	97.89±0.43 97.43±0.55	51	1.08
$\mathrm{MP}^{\mathrm{PCA(10)}}$	**97.97±0.58** **97.76±0.70**	13	97.75±0.32 97.43±0.38	81	1.24

TABLE III
CLASSIFICATION RESULTS OBTAINED FOR THE HIGH SPATIAL RESOLUTION DATA SETS. (a) WASHINGTON, DC AND (b) PAVIA DATA SETS. THE DIFFERENCE AT A LEVEL OF SIGNIFICANCE OF 5% ($Z_{ij} > 1.96$) IS HIGHLIGHTED IN BOLDFACE

(a)

Features	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
Spectral	**85.77±1.63** **81.10±0.71**	6	84.41±1.42 80.67±1.32	161	**14.64**
$\mathrm{MP}^{\mathrm{PCA(5)}}$	**95.42±0.82** **93.94±0.40**	4	95.16±0.70 93.33±0.30	31	**3.74**
$\mathrm{MP}^{\mathrm{PCA(10)}}$	**96.16±0.50** **94.78±0.36**	5	95.87±0.65 94.17±0.43	57	**4.51**

(b)

Features	DE-ELM		DE-SVM		*McNemar's* test
	OA AA	Time [S]	OA AA	Time [S]	$Z_{ELM,SVM}$
Spectral	83.57±1.53 87.69±0.88	7	**83.65±1.27** **88.23±0.85**	56	-0.34
$\mathrm{MP}^{\mathrm{PCA(5)}}$	**96.83±1.09** **97.50±0.62**	7	95.92±1.20 96.89±0.99	26	**10.60**
$\mathrm{MP}^{\mathrm{PCA(10)}}$	**98.04±0.63** **98.37±0.47**	8	97.25±0.97 97.77±0.87	39	**11.08**

test [3] to compare the ELM performances with the state-of-the-art SVM from a statistical point of view. For the SVM, we use the LIBSVM software [20] under a MATLAB environment.

As for the ELM, we implement analytical solution (9) using a few lines of codes also in MATLAB. The term $(\mathbf{I}/C + \mathbf{K})^{-1}\boldsymbol{\eta}$ is computed using the matrix division operator $(\mathbf{I}/C + \mathbf{K}) \setminus \boldsymbol{\eta}$, which produces the solution by using Gaussian elimination. The division operator "\" produces a solution that is two to three times faster than the command "*inv*". In the experiments, for both classifiers, we adopt the common Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}) = exp(-\gamma\|\mathbf{x}_i - \mathbf{x}\|^2)$, where γ represents a parameter that is inversely proportional to the width of the Gaussian kernel. For a fair comparison, we also tune the parameters of the SVM C and γ using the DE. We set the parameters of the DE as follows: population size = 10, number of function evaluation $FES_{\max} = 100$, and CR and F to the standard value of 0.9. We set the search boundary of C and γ in the ranges [10^{-3}, 1000] and [10^{-3}, 10], respectively. To compute objective function $\boldsymbol{g}(\vec{\mathbf{v}}_i)$ (i.e., the cross-validation accuracy), we set the number of folds $k = 3$.

B. Results

Classification With DE-ELM and DE-SVM: Tables II and III list the classification results obtained by both classifiers using the spectral, $\mathrm{MP}^{\mathrm{PCA(5)}}$, and $\mathrm{MP}^{\mathrm{PCA(10)}}$ features. These last two are the features generated by applying the opening and closing operations with reconstruction using a disk-shaped structural element in the range [3, 12] with a step of three to the top five and ten PCA features, respectively. This will result in 40 and 80 features for both scenarios, respectively. These tables show that the inclusion of MP features will lead to significant improvement in terms of the classification accuracy with respect to the classification method that is only based on the spectral features, confirming the findings of recent remote sensing studies.

From these results, one can see that the DE-ELM is statistically significantly better with respect to the DE-SVM in seven cases, whereas the SVM is only better in one case. The classifiers similarly perform for the remaining four cases. In terms of computation time, the results show that the DE-ELM is faster than the DE-SVM. For example, for the Indiana data set, the computation is 296, 112, and 180 s using the spectral, $\mathrm{MP}^{\mathrm{PCA(5)}}$, and $\mathrm{MP}^{\mathrm{PCA(10)}}$ features, respectively, whereas it is only equal to 18, 14, and 14 s, respectively, for the DE-ELM. This means that, for these cases, the DE-ELM is 16, 8, and 13 times faster than the DE-SVM. For the Washington DC data set, the computation time of the DE-SVM using the aforementioned features is 161, 31, and 57 s, respectively, whereas for the DE-ELM, it is equal to 6, 4, and 5 s, respectively. This means that the DE-ELM is 27, 8, and 4 times faster.

Sensitivity With Respect to the Number of Training Samples: In order to further investigate the performance of the DE-ELM, we repeated the given experiments using different training set sizes (i.e., 40, 30, 20, and 10 samples). Tables IV and V list the detailed results for the scenario $\mathrm{MP}^{\mathrm{PCA(10)}}$. Here, we again observe that among the 16 cases, the DE-ELM is statistically significantly better in 12 cases. Again, the computation time is much less than that of the DE-SVM.

IV. CONCLUSION

In this letter, we have proposed an efficient classification method for hyperspectral images based on the ELM and MP features. In particular, we have developed an automatic method to solve the model selection issue that is associated with this classifier based on the DE optimization. The experimental results obtained by the proposed DE-ELM on four hyperspectral data sets shows that, in most cases, the DE-ELM provides better classification accuracy with respect to the state-of-the-art SVM and that it is faster as its solution is simple, only requiring the inversion of a kernel matrix that is computed from the training samples.

TABLE IV
CLASSIFICATION RESULTS OBTAINED FOR THE LOW SPATIAL RESOLUTION DATA SETS. (a) AVIRIS AND (b) KSC DATA SETS. THE DIFFERENCE AT A LEVEL OF SIGNIFICANCE OF 5% ($Z_{\mathrm{ELM,SVM}} > 1.96$) IS HIGHLIGHTED IN BOLDFACE

(a)

Training Samples	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
10	**83.56±1.31 91.33±0.59**	2	82.04±2.48 90.07±1.38	25	**5.10**
20	**89.90±1.66 94.99±0.83**	4	89.10±2.49 93.97±0.94	56	**2.96**
30	**92.15±1.47 96.19±0.56**	8	91.38±1.37 95.57±0.58	90	**3.39**
40	**94.25±0.58 97.10±0.34**	11	93.22±0.97 96.45±0.46	140	**4.85**

(b)

Training Samples	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
10	**92.11±1.89 90.70±2.33**	2	89.72±1.54 87.96±1.42	13	**6.07**
20	**95.71±1.15 94.98±1.20**	3	94.90±0.65 93.67±0.75	27	**3.01**
30	**97.23±0.34 96.91±0.41**	6	96.86±0.41 96.17±0.34	44	1.65
40	**97.72±0.30 97.50±0.36**	9	97.42±0.52 97.00±0.52	61	1.48

TABLE V
CLASSIFICATION RESULTS OBTAINED FOR THE HIGH SPATIAL RESOLUTION DATA SETS. (a) HYDICE AND (b) PAVIA DATA SETS. THE DIFFERENCE AT A LEVEL OF SIGNIFICANCE OF 5% ($Z_{ij} > 1.96$) IS HIGHLIGHTED IN BOLDFACE

(a)

Training Samples	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
10	**88.46±2.70 85.64±1.71**	1	87.82±3.84 85.54±2.16	5	**4.63**
20	**91.81±2.30 90.07±0.81**	2	91.46±3.03 89.50±1.19	14	**5.04**
30	**93.89±1.96 92.28±1.32**	3	**93.71±2.72 91.73±1.54**	25	1.71
40	**95.66±0.55 93.88±0.52**	4	**95.37±0.70 93.28±0.56**	39	**4.77**

(b)

Training Samples	DE-ELM		DE-SVM		*McNemar's* test
	OA±σ_{OA} AA±σ_{AA}	Time [S]	OA±σ_{OA} AA±σ_{AA}	Time [S]	$Z_{ELM,SVM}$
10	**91.10±3.07 93.07±1.42**	2	86.97±3.80 92.04±1.67	7	**27.72**
20	**94.91±1.47 95.46±0.90**	3	92.06±3.50 94.54±1.53	15	**22.35**
30	**95.80±1.34 96.55±0.98**	4	94.21±255 95.93±1.08	24	**15.10**
40	**97.42±0.68 97.90±0.49**	5	95.99±1.31 97.04±0.71	31	**16.69**

Finally, it is worth mentioning that the model selection strategies that are based on other evolutionary computation methods, such as particle swarm optimization and genetic algorithms, could be investigated as potential alternatives to the proposed DE-based solution.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe, Prof. P. Gamba (University of Pavia, Pavia, Italy), and Prof. M. Crawford (Purdue University, West Lafayette, IN, USA) for providing the hyperspectral data sets used in the experiments.

REFERENCES

[1] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote-sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1178–1790, Aug. 2004.
[2] J. Ham, Y. Chen, M. M Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
[3] Y. Bazi and F. Melgani, "Toward an optimal SVM Classification system for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, Nov. 2006.
[4] N. Alajlan, Y. Bazi, F. Melgani, and R. R. Yager, "Fusion of supervised and unsupervised learning paradigms for improved classification of hyperspectral images," *Inf. Sci.*, vol. 217, pp. 39–55, Dec. 2012.
[5] J. Li, P. R Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.
[6] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.
[7] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.
[8] P. Gurram and H. Kwon, "Sparse kernel based ensemble learning with fully optimized kernel parameters for hyperspectral classification problems," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 787–802, Feb. 2013.
[9] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.
[10] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structural sparse logistic regression and three dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.
[11] N. Alajlan, Y. Bazi, H. AlHichri, F. Melgani, and R. R. Yager, "Using OWA operators for the classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Obser. Remote Sens.*, vol. 6, no. 2, pp. 602–614, Apr. 2013.
[12] G. B. Huang, L. Chen, and S. K. Siew, "Universal approximation using incremental constructive feedforward neural networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.
[13] G. B. Huang and L. Chen, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, Dec. 2006.
[14] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[15] J. Cao, Z. Lin, and G. B. Huang, "Self-adaptive evolutionary extreme machine learning," *Neural Process. Lett.*, vol. 36, no. 3, pp. 285–305, Dec. 2012.
[16] L. Pasoli, F. Melgani, and E. Blanzieri, "Gaussian process regression for estimating chlorophyll concentration in subsurface waters from remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 3, pp. 464–468, Jul. 2010.
[17] Y. Bazi, N. Alajlan, and F. Melgani, "Improved estimation of water chlorophyll concentration with semisupervised Gaussian process regression," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 7, pp. 2733–2743, Jul. 2012.
[18] S. Das and P. N. Suganthan, "Differential evolution: A survey of state- of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[19] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Inf. Sci.*, vol. 185, no. 1, pp. 153–177, Feb. 2012.
[20] LIBSVM—A Library for Support Vector Machines. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm