Extreme Learning Machine With Composite Kernels for Hyperspectral Image Classification

Yicong Zhou, *Senior Member, IEEE*, Jiangtao Peng, and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*—Due to its simple, fast, and good generalization ability, extreme learning machine (ELM) has recently drawn increasing attention in the pattern recognition and machine learning fields. To investigate the performance of ELM on the hyperspectral images (HSIs), this paper proposes two spatial–spectral composite kernel (CK) ELM classification methods. In the proposed CK framework, the single spatial or spectral kernel consists of activation–function-based kernel and general Gaussian kernel, respectively. The proposed methods inherit the advantages of ELM and have an analytic solution to directly implement the multiclass classification. Experimental results on three benchmark hyperspectral datasets demonstrate that the proposed ELM with CK methods outperform the general ELM, SVM, and SVM with CK methods.**

***Index Terms*—Composite kernel (CK), extreme learning machine (ELM), hyperspectral image (HSI) classification.**

I. Introduction

HYPERSPECTRAL data contain rich spectral and spatial information of the materials in a given geographic scene. In hyperspectral images (HSIs), each pixel consists of the spectral characteristics across a continuous range of narrow bands, and each image contains the scene structure information. These special characteristics bring wide applications of HSIs and also pose many processing problems [1], such as feature extraction and classification. Classification is an important task in HSI processing. Based on the fact that different materials have different reflections at a certain spectral band, the traditional pixelwise classifiers usually identify and classify the materials based on their spectral curves (pixels). Due to the high HSI dimensionality coupled with limited labeled samples [2], HSI classification usually suffers from the Hughes phenomenon [3] and becomes a challenging problem [1], [2], [4]. To overcome the high-dimensional and small-sample problem, many machine learning methods, such as support vector machine (SVM) and other kernel-based methods [5]–[7], and semi-supervised learning methods [8], were recently introduced for HSI classification and had shown good performance [5]–[10].

However, these pixelwise classifiers only use the spectral information without considering the rich spatial information [2]. In the pixelwise classification, each labeled HSI pixel is a sampled spectrum and processed independently. In fact, spatial neighboring pixels have similar spectral characteristics and usually belong to the same class. The inter-pixel correlations can be used to improve HSI classification performance [1], [2], [11]. In recent years, many spatial–spectral classification methods have been proposed [1], [11]–[14], such as the morphological transformation-based spatial–spectral classifier [14]–[17], Markov random fields (MRFs) methods [18], [19], and SVM with composite kernels (SVM-CKs) [20]–[22]. In SVM-CK, each pixel is considered as both the spectral and spatial features. The spatial feature is usually represented as the mean or standard deviation of pixels in a spatial neighborhood [20]. By integrating spectral and spatial features in a CK, SVM-CK is effective and easy to implement [20]. However, the optimal parameter of SVM-CK is usually difficult to find. Searching the optimal CK parameters and SVM regularization parameter is time-consuming, especially for the high-dimensional HSI data.

Recently, a fast and effective machine learning method called the extreme learning machine (ELM) has been proposed [23]–[26]. As a single hidden layer feedforward neural network (SLFN) [27], [28], ELM does not need to tune the hidden layer parameters if the network architecture (number of the hidden layer nodes) is determined. The hidden layer parameters in ELM are randomly generated and independent of the training data and application environments. By minimizing the training error and the norm of output weights simultaneously, ELM tends to have better generalization performance and has a unified analytic solution for the binary, multiclass, and regression problems [25]. The implementation of ELM is simple because only (regularized) least squares is involved. In addition, the general ELM can be extended to kernel learning framework. Due to these remarkable properties, ELM has been applied to different fields [25], [26], [29]–[32]. In the field of HSI processing, Pal *et al.* applied the general ELM and kernel-based ELM to land cover classification [29], [33], where ELM provides a slightly better results than the backpropagation neural network and SVM. However, the computational cost of ELM is much less than the backpropagation neural network and SVM. Bazi

Manuscript received May 01, 2014; revised June 28, 2014; accepted September 18, 2014. Date of publication October 15, 2014; date of current version July 30, 2015. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1, in part by the Research Committee at the University of Macau under Grant MYRG2014-00003-FST, Grant MRG017/ZYC/2014/FST, Grant MYRG113(Y1-L3)-FST12-ZYC, and Grant MRG001/ZYC/2013/FST, and in part by the National Natural Science Foundation of China under Grant 11371007. *(Corresponding author: Jiangtao Peng.)*

Y. Zhou and C. L. P. Chen are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@umac.mo; Philip.Chen@ieee.org).

J. Peng is with the Faculty of Mathematics and Statistics and Key Laboratory of Applied Mathematics of Hubei Province, Hubei University, Wuhan 430062, China, and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: pengjt1982@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2014.2359965

et al. used a differential evolution method to select the optimal parameters in the kernel-based ELM for HSI classification [34]. Samat *et al.* proposed two ensemble ELM methods based on Bagging and AdaBoost to improve the stability of ELM [35]. Heras *et al.* explored two ELM-based spatial–spectral classifiers for HSI classification using watershed transform and spatially regularization methods [36].

In this paper, based on ELM, we perform the joint spatial–spectral classification of HSIs. The fast computation capability of ELM can reduce the time for processing the high dimensionality and high-resolution HSI data. The good generalization performance of ELM helps to increase both the spectral and spatial separability and then leads to a superior performance by joint spatial–spectral classification. We propose two ELM-based CK spatial–spectral classification methods, namely, ELM with CK (ELM-CK) and kernel-based ELM with CK (KELM-CK). In ELM-CK, the single spatial or spectral kernel is represented as the multiplication of a random hidden layer output matrix and its transposition. While in KELM-CK, the general Gaussian radial basis function (RBF) kernel is used. The composite spatial–spectral kernel is input to the ELM learning framework, resulting in a simple and effective solution for HSI classification.

The rest of this paper is organized as follows. The ELM method is introduced in Section II. In Section III, the proposed ELM with CK methods are described. The experimental results and analysis are provided in Section IV. Finally, Section V gives a summary of our work.

II. Extreme Learning Machine

A. Single-Hidden Layer Feedforward Neural Networks

Given N training samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]^{\mathrm{T}} \in R^d$ and $\mathbf{y}_i = [y_{i1}, \ldots, y_{im}]^{\mathrm{T}} \in R^m$, the output function of a standard SLFNs with L hidden nodes can be represented as

$$\boldsymbol{f}_L(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G_i(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\boldsymbol{a}_i, b_i, \mathbf{x}) \tag{1}$$

where $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{id}]^{\mathrm{T}}$ is the weight vector connecting the input nodes to the ith hidden node, $\boldsymbol{\beta}_i = [\beta_{i1}, \ldots, \beta_{im}]^{\mathrm{T}}$ is the weight vector connecting the ith hidden node to the output nodes, b_i is the threshold of the ith hidden node, and $G_i(\mathbf{x}) = G(\boldsymbol{a}_i, b_i, \mathbf{x})$ is the hidden layer output function (activation function) of node i.

The above SLFNs with L hidden nodes and activation function G can approximate N arbitrary distinct samples [23]: $\sum_{j=1}^{N} \|\boldsymbol{f}_L(\mathbf{x}_j) - \mathbf{y}_j\| = 0$, i.e., there exist $\boldsymbol{\beta}_i$, $\boldsymbol{a}_i$, and b_i such that

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i G(\boldsymbol{a}_i, b_i, \mathbf{x}_j) = \mathbf{y}_j, \quad j = 1, \ldots, N. \tag{2}$$

The above N equations can be written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \tag{3}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_L]^{\mathrm{T}} \in R^{L\times m}$ and $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_N]^{\mathrm{T}} \in R^{N\times m}$, and the hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} G(\boldsymbol{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\boldsymbol{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ G(\boldsymbol{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\boldsymbol{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}. \tag{4}$$

The matrix $\mathbf{H}$ is a function of hidden layer parameters $\boldsymbol{a}_i$ and b_i and is unknown. During the training process, SLFN needs to simultaneously adjust the parameters $\boldsymbol{a}_i$ and b_i, $\boldsymbol{\beta}_i$ such that

$$(\hat{\boldsymbol{a}}_i, \hat{b}_i, \hat{\boldsymbol{\beta}}) = \arg\min_{\boldsymbol{a}_i, b_i, \boldsymbol{\beta}} \|\mathbf{H}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L; b_1, \ldots, b_L)\boldsymbol{\beta} - \mathbf{Y}\|^2. \tag{5}$$

Equation (5) is usually solved by gradient-based iterative techniques, such as back-propagation (BP) algorithm. However, the BP algorithm usually suffers from several issues such as slow learning speed, trivial parameter tuning, local minima, and over-fitting [23].

B. Extreme Learning Machine

ELM is a generalized SLFN [23], [26]. Different from the traditional SLFNs, ELM does not need to tune the hidden parameters $\boldsymbol{a}_i$ and b_i. In ELM, the weight vector $\boldsymbol{a}_i$ and threshold b_i are randomly generated in the beginning of learning and remain fixed during the learning process. Once the random values of $\boldsymbol{a}_i$ and b_i are assigned, the hidden layer output matrix $\mathbf{H}$ also keeps unchanged in the learning process [23].

Different from traditional gradient-based learning algorithms, ELM aims to reach not only the smallest training error as in (5) but also the smallest norm of output weights [25]

$$\text{Minimize}: \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2 \quad \text{and} \quad \|\boldsymbol{\beta}\|^2. \tag{6}$$

According to Bartlett's neural network generalization theory [37], for feedforward neural networks reaching smaller training error, the smaller the norms of weights are, the better generalization performance the networks tend to have. Thus, ELM tends to have better performance in real applications [25].

Denote $\mathbf{h}(\mathbf{x}) = [G(\boldsymbol{a}_1, b_1, \mathbf{x}), \ldots, G(\boldsymbol{a}_L, b_L, \mathbf{x})]$, from the optimization theory point of view, (6) can be reformulated as

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\frac{1}{2}\sum_{i=1}^{N} \boldsymbol{\xi}_i^2 \\ &\text{s.t.} \quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i^{\mathrm{T}} - \boldsymbol{\xi}_i^{\mathrm{T}}, \ i = 1, \ldots, N \end{aligned} \tag{7}$$

where $\boldsymbol{\xi}_i$ is the training error of training sample $\mathbf{x}_i$ and C is a regularization parameter.

Based on the Lagrange multiplier method and from the KKT optimality conditions [25], the solution of (7) can be analytically expressed as

$$\boldsymbol{\beta} = \mathbf{H}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{Y}. \tag{8}$$

When $\boldsymbol{\beta}$ is obtained, the output function of ELM is

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} = \mathbf{h}(\mathbf{x})\mathbf{H}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{Y}. \tag{9}$$

Similar to SVM, the general ELM in (9) can be generalized to kernel version using a kernel trick. In detail, the inner product operation involved in the computation of $\mathbf{h}(\mathbf{x})\mathbf{H}^{\mathrm{T}}$ and $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ can be replaced by a kernel function: $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$. After the substitution, we can obtain the kernel ELM (KELM) with the output function

$$\mathbf{f}(\mathbf{x}) = \mathbf{K}_{\mathbf{x}} \left(\frac{\mathbf{I}}{C} + \mathbf{K} \right)^{-1} \mathbf{Y} \tag{10}$$

where the kernel $\mathbf{K} = [\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{N}$ and $\mathbf{K}_{\mathbf{x}} = [\mathbf{K}(\mathbf{x}, \mathbf{x}_1), \ldots, \mathbf{K}(\mathbf{x}, \mathbf{x}_N)]$. The characteristic of KELM is that the general kernel functions used in SVM and other kernel-based methods can also be used in (10). In the experiment, it only needs to provide a kernel function and then to tune the related kernel parameters.

Now, we have obtained two implementations of ELM: 1) general ELM in (9) and 2) KELM in (10). The difference between ELM and KELM lies in that the output function in (9) is obtained based on the matrix $\mathbf{H}$, while the output function in (10) is computed from $\mathbf{K}$. $\mathbf{H}$ is the hidden layer output matrix defined in (4) and is dependent on the network architecture. However, $\mathbf{K}$ is independent on the network architecture and can be chosen as any kernel function. On the other hand, the output function $\mathbf{f}$ can be considered as a linear expansion over given basis functions. In this sense, ELM and KELM have different basis functions, i.e., $\mathbf{h}(\mathbf{x})$ and $\mathbf{K}_{\mathbf{x}}$, and different expansion coefficients, i.e., $\boldsymbol{\beta}$ and $\boldsymbol{\alpha} = \left(\frac{\mathbf{I}}{C} + \mathbf{K}\right)^{-1} \mathbf{Y}$, respectively. In the implementation of ELM, it needs to solve the coefficient $\boldsymbol{\beta}$ defined in (8). While in KELM, it needs to solve the coefficient $\boldsymbol{\alpha}$.

In the general ELM, the activation function $G(\mathbf{x})$ in the feature mapping $\mathbf{h}(\mathbf{x})$ can be set as a nonlinear piecewise continuous function satisfying the ELM universal approximation capability theorems [23], [25], such as Sigmoid and Gaussian functions. The Sigmoid function $G(\boldsymbol{a}, b, \mathbf{x}) = 1/(1 + \exp(-(\boldsymbol{a} \cdot \mathbf{x}) + b))$ is used in this paper. The hidden parameters $\boldsymbol{a}$ and b in the Sigmoid function are randomly generated from a uniform distribution before seeing the training data. That is, the hidden parameters $\boldsymbol{a}$ and b in ELM are not only independent of the training data but also independent of each other. In addition, we can see from (8) that when the number of training samples is very small, the size of matrix $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is very small, so ELM is very fast. When the number of training samples is huge, the ELM solution in (8) can be changed to be [25]: $\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \left(\frac{\mathbf{I}}{C} + \mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1} \mathbf{H}^{\mathrm{T}}\mathbf{Y}$. In this case, the computation cost of ELM is related to the number of hidden neurons L, which is usually not larger than 1000 [25]. Thus, ELM also achieves a low computational cost. Furthermore, ELM is simple and much effective. The good generalization performance of ELM lies in its universal approximation capability [24], [25] and classification capability [25]. The above advantages of ELM can benefit the HSI classification. With the randomly generated parameters $\boldsymbol{a}$ and b, ELM can overcome the trivial parameter tuning and overfitting problem. The low computational cost of ELM helps to increase the speed of high dimensional HSI data processing.

III. Proposed ELM With CKs

According to the spatial homogeneous distribution characteristics of HSI, a pixel and its spatial neighboring pixels generally belong to the same material and have the same label. These spatial interpixel correlations can be considered and combined with the spectral similarity to produce a desired joint spatial–spectral classifier. By exploiting the complementary discriminant information in spatial-domain and spectral-domain, the CK method is commonly used to perform the spatial–spectral classification. In the CK method, a local spatial feature extraction method is firstly used to extract spatial features, then the extracted spatial features and the original spectral features are used to compute spatial and spectral kernels, which are combined to form a CK. Based on the computed CK, the kernel-based methods can be used to perform the final classification. In this section, we use ELM to perform the CK-based spatial–spectral classification of HSIs.

Now, we extract the spectral and spatial characteristics of HSIs. Given a pixel $\mathbf{x}_i$ (here, a pixel is a sample consisting of the spectral characteristics across a continuous range of spectral bands), we denote its spectral and spatial features as $\mathbf{x}_i^{\omega}$ and $\mathbf{x}_i^{s}$, respectively. The spectral feature vector $\mathbf{x}_i^{\omega}$ is the original $\mathbf{x}_i$, which consists of spectral reflection values across all bands. The spatial feature vector $\mathbf{x}_i^{s}$ is extracted from the local spatial neighborhood of pixel $\mathbf{x}_i$ and defined as the mean of pixels in the spatial neighborhood of $\mathbf{x}_i$ in this paper.

Once the spatial and spectral features $\mathbf{x}_i^{s}$ and $\mathbf{x}_i^{\omega}$ are constructed, we can compute the corresponding ELM hidden layer output matrices $\mathbf{H}_s$ and $\mathbf{H}_{\omega}$, respectively. As in the general ELM framework of (9), we can obtain $\mathbf{K}_{\mathbf{H}_s} = \mathbf{H}_s\mathbf{H}_s^{\mathrm{T}}$ and $\mathbf{K}_{\mathbf{H}_{\omega}} = \mathbf{H}_{\omega}\mathbf{H}_{\omega}^{\mathrm{T}}$ and denote them as the spatial and spectral activation–function-based kernels, respectively. Using both the spectral and spatial information, ELM with CK (ELM-CK) can be represented as

$$\mathbf{K} = \mu\mathbf{K}_{\mathbf{H}_s} + (1-\mu)\mathbf{K}_{\mathbf{H}_{\omega}} = \mu\mathbf{H}_s\mathbf{H}_s^{\mathrm{T}} + (1-\mu)\mathbf{H}_{\omega}\mathbf{H}_{\omega}^{\mathrm{T}} \tag{11}$$

where μ is a combination coefficient balancing the spatial and spectral information.

For KELM in (10), we can compute the spatial kernel $\mathbf{K}_s$ and spectral kernel $\mathbf{K}_{\omega}$ as follows:

$$\mathbf{K}_s(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i^{s} - \mathbf{x}_j^{s}\|^2}{2\sigma_s^2}\right) \tag{12}$$

$$\mathbf{K}_{\omega}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i^{\omega} - \mathbf{x}_j^{\omega}\|^2}{2\sigma_{\omega}^2}\right) \tag{13}$$

where the RBF kernel is used, and σ_s and σ_{ω} are the widths of the spatial and spectral RBF kernels. KELM with CK (KELM-CK) is represented as

$$\mathbf{K} = \mu\mathbf{K}_s + (1-\mu)\mathbf{K}_{\omega}. \tag{14}$$

When the composite spatial–spectral kernel in (11) or (14) is computed, the ELM model resolves a coefficient

$$\boldsymbol{\alpha} = \left(\frac{\mathbf{I}}{C} + \mathbf{K}\right)^{-1} \mathbf{Y} \tag{15}$$

and outputs

$$\mathbf{f}(\mathbf{x}) = \mathbf{K}_{\mathbf{x}}\boldsymbol{\alpha} \triangleq [\mathbf{f}_1(\mathbf{x}), \ldots, \mathbf{f}_m(\mathbf{x})]. \tag{16}$$

During the prediction phase, each test sample $\mathbf{x}_t$ will be assigned to the index corresponding to the highest value in $\mathbf{f}(\mathbf{x}_t) = [\mathbf{f}_1(\mathbf{x}_t), \ldots, \mathbf{f}_m(\mathbf{x}_t)]$.

In the proposed methods, we should note the following.

1) ELM-CK inherits the computational advantages of ELM: simple and fast. It does not need to tune the hidden node parameters $\boldsymbol{a}_i$ and b_i in the activation–function-based kernel $\mathbf{K}_{\mathbf{H}_s}$ and $\mathbf{K}_{\mathbf{H}_\omega}$. In addition, it is computationally effective and extremely fast because the kernel size $N \times N$ is small in HSI classification, where only limited labeled samples are available.
2) KELM-CK has the same CK (14) with SVM-CK. However, the solving procedure for the coefficient vector $\boldsymbol{\alpha}$ in KELM-CK is different from that in SVM-CK, especially for the multiclass classification. SVM implements the multiclass classification by means of binary classification according to the one-against-all (OAA) or one-against-one (OAO) strategy. However, ELM can directly perform multiclass classification using multioutput nodes [25]. ELM has a unified solution in (15) for the multiclass and binary classification problems.
3) For the spatial local neighborhood feature extraction, a mean feature is used. Because the neighboring pixels are similar, the mean statistics can measure the central tendency and reflect the overall characteristics of neighboring pixels. For the spectral samples belonging to different materials (different homogeneous regions), their spatial neighboring information will be different. The spatial variability measured by the mean pixel can be considered as a complementarity of the spectral signature to improve the discriminant ability. Besides the mean statistics [20], [21], the weighted mean, median [22], standard deviation [20], nonlinear (weighted) mean statistics [12], [38], and morphological operators [39], [40] can also be used to extract the local spatial features.
4) In (11) and (14), the weighted summation CK is used. Other CKs, such as stacked kernel, direct summation kernel, and cross-information kernel [20], [40], can also be used. In ELM-CK in (11), the activation–function-based kernel is represented as the multiplication of a random hidden layer output matrix and its transposition, and the combination of the spatial and spectral activation–function-based kernels is first used in this paper. In KELM-CK in (14), the spatial or spectral kernel is represented as the commonly used RBF kernel due to its universal approximating property [41] and asymptotic behaviors for both linear and nonlinear classification [42].
5) Although both methods are spatial–spectral classifiers, the proposed ELM-CK is different from Hares's watershed-based ELM algorithm. In the watershed-based ELM, it first needs to compute a robust color morphological gradient (RCMG)-based one-band gradient image from the original multiband HSIs. Then a watershed algorithm is performed on the one-band gradient image to generate a segmentation map. Finally, the segmentation map and spectral pixelwise ELM map are combined using a majority voting process. While in ELM-CK, it only needs to extract local spatial features by computing the mean of neighboring pixels, and sums the spatial kernel and spectral kernel for the standard ELM classification. So, ELM-CK is much simpler than watershed-based ELM. In addition, watershed-based ELM needs more information than ELM-CK, because it needs to process the whole HSI scenes to obtain a spatial segmentation map, while ELM-CK only needs the spatial neighborhood information for each pixel. That is, watershed-based ELM needs to see the whole HSI data before training, while ELM-CK only needs spatial neighbors of training pixels.

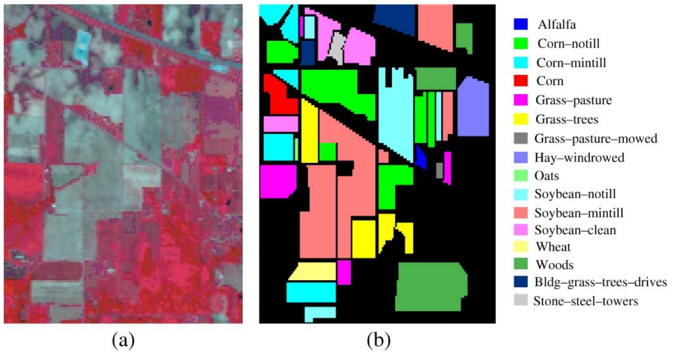


Fig. 1. Indian Pines dataset. (a) RGB composite image of three bands 50, 27, and 17. (b) Ground-truth map.

IV. Experimental Results

A. Hyperspectral Datasets

In this section, we investigate the performance of the proposed ELM-CK and KELM-CK on three benchmark hyperspectral datasets, i.e., Indian Pines, Univerisy of Pavia, and Salinas. The three datasets are public available hyperspectral datasets.[1] The Indian Pines dataset is a classic benchmark to evaluate the performance of HSI classification algorithms because of the widespread presence of mixed pixels in all available classes and unbalanced number of labeled pixels per class. The Univerisy of Pavia and Salinas datasets have very high spatial resolution and large number of labeled pixels.

1) Indian Pines: The dataset was acquired by the AVIRIS sensor in 1992. The image scene contains 145×145 pixels and 220 spectral bands, where 20 channels were discarded because of the atmospheric affection. The spatial resolution of the data is 20 m per pixel. There are 16 classes and totally 10 249 labeled samples in the dataset. The false color composition of bands 50, 27, and 17 and the ground-truth map are shown in Fig. 1.

2) University of Pavia: This dataset was acquired in 2001 by the ROSIS instrument over the city of Pavia, Italy. This image scene corresponds to the University of Pavia and has the size of 610×340 pixels and 115 spectral bands. The spatial resolution is 1.3 m per pixel. After discarding noisy and

[1] Available online: http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

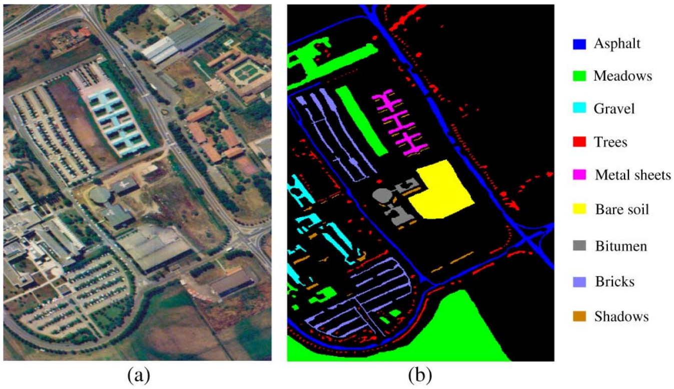


Fig. 2. University of Pavia dataset. (a) RGB composite image of three bands 60, 30, and 2. (b) Ground-truth map.

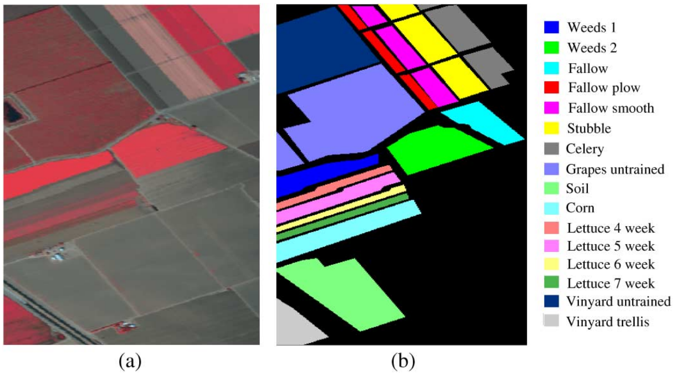


Fig. 3. Salinas dataset. (a) RGB composite image of three bands 50, 30, and 20. (b) Ground-truth map.

water absorption bands, 103 bands are retained. The data contain 9 ground-truth classes and 42 776 labeled samples in total. The false color composition of bands 60, 30, and 2 and the ground-truth map are shown in Fig. 2.

3) Salinas: The dataset was collected by the AVIRIS sensor over Salinas Valley, California, USA. It contains 512 × 217 pixels and 224 spectral bands. After discarding noisy and water absorption bands, 204 bands are retained. The spatial resolution of the data is 3.7 m per pixel. The data contain 16 ground-truth classes and 54 129 labeled samples in total. The false color composition of bands 50, 30, and 20 and the ground-truth map are shown in Fig. 3.

B. Competing Methods and Parameter Setting

The proposed ELM with CK methods (ELM-CK and KELM-CK) are compared with the classical classifiers, such as SVM, SVM with CK (SVM-CK), general ELM, and kernel-based ELM (KELM). The classification performance of different algorithms is assessed on the testing set by the overall accuracy (OA) which is the number of correctly classified testing samples divided by the number of total testing samples, by the average accuracy (AA), which represents the average of the classification accuracies for the individual classes, and by the kappa (κ) coefficient, which measures the degree of classification agreement. All experiments are conducted using MATLAB R2011b on a computer with 2.93 GHz CPU and 8.0 GB RAM. All data are normalized to have a unit ℓ_2 norm.

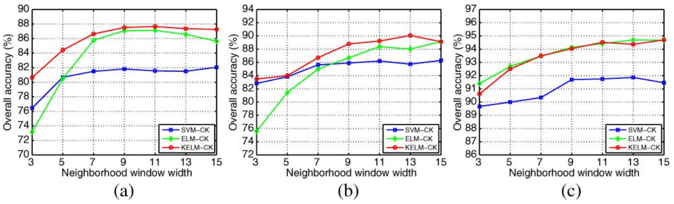


Fig. 4. OA versus neighborhood window width for three datasets. (a) Indian Pines. (b) University of Pavia. (c) Salinas.

In the general ELM method, the sigmoid function is used, the hidden layer parameters $(\mathbf{a}_i, b_i)_{i=1}^{L}$ are randomly generated based on uniform distribution from the range $[-1, 1]$, and the number of hidden nodes L is set as 1000 as recommended in [25]. For the CK methods, SVM-CK, ELM-CK, and KELM-CK, the combination coefficient μ is set to be 0.8. For all kernel-based algorithms, the Gaussian RBF kernel is used. The LIBSVM software under a MATLAB environment is used for the implementation of SVM methods [43]. The RBF kernel parameter σ and penalty parameter C involved in these methods are needed to be tuned. The parameter σ varies in the range $\{2^{-4}, 2^{-3}, \ldots, 2^{4}\}$, and C ranges from 10^0 to 10^5. We use a threefold cross-validation with a grid search method to select the optimal parameters. In detail, for each parameter pair (C, σ) (or C for ELM and ELM-CK, or $(C, \sigma_s, \sigma_\omega)$ for SVM-CK and KELM-CK), it performs the following cross-validation operations: 1) the original training set is randomly divided into three equally sized subsets; 2) for the three subsets, two subsets are used to train the model and the remaining subset is used as the validation data for testing the model and outputting OA; 3) the Step 2) is repeated three times (folds) such that each of the three subsets are used as the validation data once; and 4) the three results from the folds are averaged to produce a single OA. Finally, the parameter pair with the highest OA obtained by the cross-validation process is set as the optimal parameter pair. The optimal parameter pair corresponds to the highest empirically cross-validation OA and is used for training and testing.

C. Investigation on the Effect of Neighborhood Window Size

The neighborhood window size decides the spatial local pixel neighborhood and hence affects the performance of spatial–spectral classifiers. Here, we investigate the OA values of SVM-CK, ELM-CK, and KELM-CK on seven different window widthes, i.e., 3, 5, 7, 9, 11, 13, and 15. Twenty samples in each class are selected as the training set, and the remaining samples are set as testing samples. Fig. 4 shows the changes in OA as a function of the window width for three datasets.

From Fig. 4, for the Indian Pines dataset, we can see that the best OAs are roughly achieved at the window width 9 for SVM-CK, ELM-CK, and KELM-CK algorithms. When the window width is smaller than 7, the three methods show bad results because the neighborhood information is insufficient to reflect spatial variability. When the window width is larger than 9, OAs are relatively stable, mainly due to the presence of large homogeneous classes as shown in Fig. 1(b). For the University of Pavia dataset, SVM-CK provides stable results when the window width is no less than 7, while ELM-CK and KELM-CK achieve relatively stable results with the need of the window width 11 at least. For the Salinas dataset, it needs the neighborhood window width at least 9 for three algorithms to obtain good results. In general, the large window benefits HSI classification for the proposed algorithms mainly due to the large homogeneous region distribution of HSIs. For consistency, we set the spatial neighborhood window width as 9 for the three datasets in the following experiments.

It can also be seen from Fig. 4 that ELM-CK provides bad results in the case of small windows for the Indian Pines and University of Pavia datasets, while KELM-CK shows consistent good results in the cases of seven different windows. This may be because the kernel function in KELM is more discriminant than the activation function in ELM especially when the spatial information is insufficient in the case of small windows. Compared with SVM-CK, KELM-CK shows consistent better results in the cases of different windows and different datasets. This verifies the good generalization performance of the ELM solution.

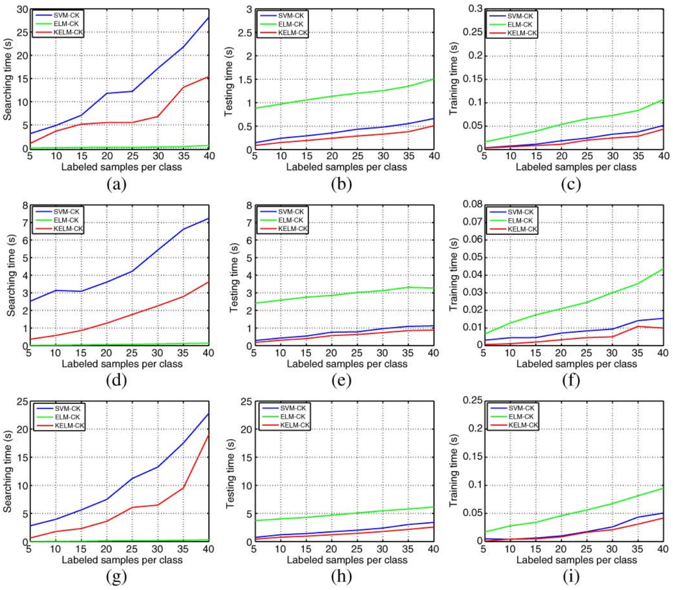


Fig. 5. Searching (left), testing (middle), and training time (right) of SVM-CK, ELM-CK, and KELM-CK for three datasets. (a)–(c) Indian Pines. (d)–(f) University of Pavia. (g)–(i) Salinas.

D. Investigation on the Computation Cost

In this section, we investigate the computation time of the proposed methods. Fig. 5 shows the searching time, testing time, and training time of the spatial–spectral SVM-CK, ELM-CK, and KELM-CK in the cases of different numbers of labeled samples per class ranging from 5 to 40 and different datasets. The searching time refers to the time used for the parameter selection. For SVM-CK and KELM-CK, there are three parameters: 1) the penalty parameter C, 2) spatial RBF kernel parameter, and 3) spectral RBF kernel parameter. In ELM-CK, only the penalty parameter C needs to be tuned.

It is noticeable from Fig. 5 that the searching (or training, testing) time curves are similar on three datasets. ELM-CK and KELM-CK are much faster than SVM-CK in the process of parameters searching. In the steps of training and testing, KELM-CK is slightly faster than SVM-CK, ELM-CK is the slowest because the computation of $\mathbf{H}$ and $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ in ELM-CK is slower than the computation of $\mathbf{K}$ in SVM-CK and KELM-CK. In the whole process, the searching time is dominant, and the total time costs of ELM-CK and KELM-CK are much less than that of SVM-CK.

TABLE I
CLASSIFICATION ACCURACIES (%) WITH 5% LABELED SAMPLES PER CLASS FOR THE INDIAN PINES DATASET

No.	# Samples		Spectral classifiers			Spatial–spectral classifiers		
	Train	Test	SVM	ELM	KELM	SVM-CK	ELM-CK	KELM-CK
1	3	43	42.6±19.0	1.16±1.98	29.3±14.3	62.1±25.7	79.3±19.9	**81.6±20.6**
2	71	1357	73.1±4.92	66.6±3.51	70.7±3.12	90.8±1.74	93.6±1.94	**94.9±2.11**
3	42	788	62.2±4.44	39.6±4.01	57.0±6.20	91.4±2.97	94.2±2.80	**95.6±2.59**
4	12	225	37.2±8.62	8.89±4.35	31.9±8.38	84.9±7.81	86.7±7.40	**92.0±4.50**
5	24	459	89.1±2.05	81.2±3.59	86.6±2.35	91.0±4.28	93.5±4.13	**93.6±3.92**
6	37	693	94.0±1.90	91.1±2.84	94.8±1.74	98.4±0.70	98.0±0.93	**99.2±0.49**
7	3	25	82.4±9.47	9.60±10.2	71.6±7.41	95.6±4.79	96.0±6.53	**96.8±5.27**
8	24	454	96.4±2.26	95.4±2.68	97.4±1.40	98.1±1.59	**99.9±0.23**	99.3±0.90
9	3	17	56.5±19.1	10.6±11.7	52.9±17.8	82.9±19.1	82.9±18.2	**98.8±3.72**
10	49	923	64.9±5.80	42.9±6.48	65.5±6.65	85.5±6.35	89.2±6.07	**90.1±4.29**
11	123	2332	76.5±4.47	77.6±2.60	81.3±1.77	92.9±2.18	96.7±1.79	**97.1±1.74**
12	30	563	59.2±6.51	34.2±9.18	59.7±6.56	88.5±3.32	**94.4±3.20**	92.7±3.62
13	10	195	90.8±7.77	95.8±1.98	95.5±3.48	97.9±1.93	97.9±1.80	**98.6±1.47**
14	63	1202	92.6±2.71	94.3±2.87	95.4±2.11	98.4±0.96	**99.7±0.18**	99.5±0.50
15	19	367	40.9±6.55	34.1±5.87	43.7±7.59	90.3±7.09	94.8±2.00	**95.5±2.93**
16	5	88	89.8±4.29	59.7±18.5	79.7±8.25	90.7±5.79	88.8±5.32	**90.8±9.50**
OA			75.5±0.92	67.6±1.05	76.2±1.41	92.4±0.77	95.2±0.40	**95.9±0.40**
AA			71.8±1.91	52.7±1.79	69.6±2.59	90.0±2.54	92.8±1.98	**94.8±1.44**
κ			72.0±1.05	62.3±1.31	72.7±1.69	91.3±0.88	94.6±0.45	**95.3±0.46**

E. Comparison Results

The Indian Pines dataset has an unbalanced number of labeled pixels per class, where the total number of samples ranges from 20 to 2455 in each class. Due to the unbalanced data distribution, the classification of Indian Pines dataset is a challenging problem. To investigate the performance of different algorithms in this challenging case, we randomly choose 5% of the labeled samples per class for training (for the class with extremely limited training samples, at a minimum three samples are chosen as training samples, resulting 518 training samples totally). The remaining labeled samples are used for testing. In this case, the ratio of the labeled samples to the total samples in each class is balanced. The proposed ELM-CK and KELM-CK are compared with SVM, SVM-CK, ELM, and KELM. The classification performance is measured by the class accuracy (CA), OA, AA, and kappa coefficient (κ) on the testing set. The mean and standard derivation of the classification results over 10 randomly runs are recorded in Table I, where the number of training and testing samples for each class are also included.

From Table I, we can see that, among the spectral classifiers, KELM shows slightly better results than SVM, and ELM provides the worst results especially for the classes with extremely limited training samples. This demonstrates the kernel used in KELM or SVM is more powerful than the randomly generated activation function used in ELM in the case of small

TABLE II
CLASSIFICATION ACCURACIES (%) UNDER DIFFERENT NUMBERS OF LABELED SAMPLES PER CLASS FOR THE INDIAN PINES DATASET

M	Index	Spectral classifiers			Spatial–spectral classifiers		
		SVM	ELM	KELM	SVM-CK	ELM-CK	KELM-CK
5	OA	45.1±4.88	41.3±3.28	47.3±3.55	59.0±2.71	64.5±3.81	**66.8±3.68**
	AA	59.4±2.90	52.2±2.67	60.0±3.98	73.7±1.87	78.0±2.13	**79.3±3.68**
	κ	39.0±4.71	34.8±3.10	41.3±3.54	54.3±3.09	60.3±4.07	**62.8±4.21**
10	OA	54.0±3.48	46.4±2.55	55.4±3.16	71.4±3.07	76.4±4.15	**78.5±4.41**
	AA	67.0±2.29	58.7±1.76	67.6±2.27	83.2±1.25	86.7±2.50	**88.4±2.73**
	κ	48.6±3.69	40.3±2.50	50.1±3.35	67.8±3.32	73.4±4.69	**75.8±4.95**
15	OA	59.2±2.17	52.7±3.01	61.7±1.76	76.8±2.21	82.7±3.41	**84.4±1.58**
	AA	71.1±1.98	63.7±2.74	72.6±1.15	86.7±1.21	90.5±2.18	**91.6±0.97**
	κ	54.3±2.31	47.0±3.20	57.1±1.91	73.8±2.49	80.4±3.83	**82.4±1.81**
20	OA	61.3±1.70	56.0±1.22	64.4±1.86	81.1±1.91	87.6±1.38	**87.8±1.50**
	AA	73.3±1.14	68.3±1.29	75.4±1.26	89.6±0.85	93.6±0.88	**93.8±0.73**
	κ	56.6±1.75	50.8±1.26	60.0±1.92	78.6±2.10	86.0±1.54	**86.2±1.67**
25	OA	65.1±2.19	58.1±2.18	67.0±1.82	83.9±2.22	89.7±1.19	**89.8±1.09**
	AA	75.7±1.76	70.4±1.97	77.1±1.25	91.2±1.42	94.8±0.61	**94.8±0.57**
	κ	60.8±2.40	53.2±2.24	62.8±2.00	81.8±2.49	88.3±1.33	**88.4±1.21**
30	OA	67.1±2.05	61.6±1.62	69.9±1.98	84.7±1.79	90.4±1.15	**90.6±1.10**
	AA	77.7±1.70	72.5±2.16	79.6±1.89	92.3±1.00	95.4±0.51	**95.6±0.42**
	κ	63.0±2.24	57.0±1.73	66.2±2.16	82.6±1.990	89.1±1.28	**89.3±1.23**
35	OA	68.1±2.24	63.1±1.97	70.2±1.95	87.6±1.20	91.3±1.14	**91.7±1.54**
	AA	78.7±0.77	73.7±1.79	80.0±1.28	93.6±0.56	95.8±0.43	**96.1±0.66**
	κ	64.1±2.41	58.6±2.03	66.5±2.11	85.9±1.35	90.1±1.28	**90.6±1.74**
40	OA	70.1±1.43	64.5±0.61	71.9±1.58	89.1±0.82	92.5±1.14	**93.4±0.99**
	AA	78.8±1.60	75.2±1.76	80.8±1.92	94.4±0.67	96.3±0.49	**96.7±0.58**
	κ	66.3±1.54	60.1±0.52	68.3±1.73	87.6±0.92	91.4±1.28	**92.4±1.13**

TABLE III
CLASSIFICATION ACCURACIES (%) UNDER DIFFERENT NUMBERS OF LABELED SAMPLES PER CLASS FOR THE UNIVERSITY OF PAVIA DATASET

M	Index	Spectral classifiers			Spatial–spectral classifiers		
		SVM	ELM	KELM	SVM-CK	ELM-CK	KELM-CK
5	OA	59.2±6.17	59.6±3.66	58.1±5.83	64.4±4.54	**71.2±3.64**	68.8±7.26
	AA	70.4±3.02	65.5±3.05	69.3±3.02	72.7±3.38	**74.6±2.44**	74.2±4.88
	κ	49.8±6.08	49.4±3.75	48.5±5.51	55.5±5.20	**63.4±4.21**	60.9±8.40
10	OA	67.0±6.08	61.1±5.67	67.3±5.89	77.0±6.35	78.7±5.77	**81.2±5.40**
	AA	75.6±1.60	67.2±2.64	75.0±1.82	82.5±3.06	82.2±3.71	**83.2±4.45**
	κ	59.0±6.27	51.5±5.66	59.1±6.19	71.0±7.59	72.8±6.92	**75.8±6.68**
15	OA	68.3±4.49	63.2±2.59	66.8±3.19	82.1±4.94	84.0±2.10	**85.8±2.91**
	AA	77.9±1.43	71.3±1.02	76.6±1.58	86.0±2.49	85.4±1.77	**86.5±1.58**
	κ	60.4±4.87	53.8±2.46	58.6±3.42	77.1±5.99	79.3±2.61	**81.5±3.56**
20	OA	71.8±3.89	64.4±2.89	71.3±3.15	85.0±5.06	85.8±1.44	**87.9±2.34**
	AA	79.6±1.05	72.0±1.33	78.8±1.55	87.2±1.79	87.6±0.86	**88.0±1.61**
	κ	64.5±4.16	55.3±3.15	63.8±3.43	80.6±6.02	81.6±1.77	**84.2±2.94**
25	OA	74.5±2.99	65.1±3.28	73.3±3.28	89.7±1.81	88.3±1.54	**89.8±3.75**
	AA	81.0±1.07	72.6±1.19	80.5±1.12	90.3±1.45	88.9±0.90	**89.7±2.30**
	κ	67.5±3.25	56.1±3.21	66.1±3.68	86.5±2.33	84.7±1.89	**86.6±4.72**
30	OA	75.5±4.32	66.4±2.16	75.0±2.96	89.5±2.89	88.9±1.74	**91.7±1.34**
	AA	81.8±1.67	73.2±0.51	81.2±1.57	90.6±1.43	89.4±0.93	**91.2±1.08**
	κ	68.8±4.90	57.6±2.32	68.1±3.45	86.3±3.60	85.5±2.20	**89.1±1.73**
35	OA	76.5±2.67	66.1±2.06	75.2±3.78	90.3±1.51	89.7±1.27	**92.5±1.28**
	AA	82.6±1.07	74.0±0.77	82.3±1.39	91.1±0.94	90.8±0.49	**92.2±0.86**
	κ	70.1±3.04	57.6±2.03	68.6±4.27	87.3±1.91	86.6±1.59	**90.1±1.65**
40	OA	78.5±3.29	66.0±2.96	77.9±1.93	91.4±1.40	91.0±1.78	**93.5±1.37**
	AA	83.6±1.07	74.4±0.91	83.1±0.72	92.0±0.99	91.3±0.76	**92.7±1.13**
	κ	72.5±3.70	57.5±3.14	71.7±2.19	88.7±1.78	88.1±2.26	**91.4±1.77**

TABLE IV
CLASSIFICATION ACCURACIES (%) UNDER DIFFERENT NUMBERS OF LABELED SAMPLES PER CLASS FOR THE SALINAS DATASET

M	Index	Spectral classifiers			Spatial–spectral classifiers		
		SVM	ELM	KELM	SVM-CK	ELM-CK	KELM-CK
5	OA	80.5±3.21	81.7±2.54	83.0±3.85	82.3±3.56	**91.4±0.66**	89.9±1.14
	AA	87.7±2.32	86.8±1.71	89.4±2.14	88.6±1.75	**95.3±0.49**	93.5±1.20
	κ	78.3±3.50	79.6±2.79	81.1±4.25	80.3±3.94	**90.4±0.74**	88.7±1.25
10	OA	85.1±1.64	83.2±2.54	86.0±1.22	87.3±1.34	**92.5±0.69**	92.3±0.97
	AA	91.4±0.53	88.9±1.30	92.3±0.46	92.6±1.02	**96.3±0.53**	95.9±0.70
	κ	83.4±1.80	81.3±2.76	84.5±1.33	85.9±1.50	**91.6±0.77**	91.4±1.07
15	OA	86.4±1.66	85.4±1.92	87.6±1.25	89.5±1.71	**93.0±1.07**	92.8±1.25
	AA	92.4±0.79	91.0±0.76	93.6±0.44	94.0±0.88	**96.8±0.40**	96.6±0.46
	κ	84.9±1.83	83.7±2.09	86.2±1.36	88.4±1.89	**92.2±1.19**	92.0±1.38
20	OA	87.4±1.67	86.4±0.96	87.6±1.63	91.6±1.49	**93.9±0.63**	93.7±0.85
	AA	93.3±0.57	91.8±0.35	93.6±0.50	95.3±0.75	**97.3±0.36**	97.0±0.56
	κ	86.0±1.83	84.9±1.06	86.2±1.80	90.7±1.64	**93.2±0.70**	93.0±0.95
25	OA	88.0±1.38	86.5±0.66	87.8±1.13	92.7±0.82	93.9±1.12	**94.4±0.92**
	AA	93.7±0.56	92.5±0.49	94.1±0.56	96.3±0.56	97.3±0.54	**97.5±0.38**
	κ	86.7±1.52	85.0±0.73	86.5±1.26	91.9±0.91	93.3±1.25	**93.7±1.02**
30	OA	88.0±1.61	87.3±1.01	88.6±0.80	93.3±1.13	94.6±0.52	**95.3±1.22**
	AA	94.0±0.61	93.0±0.40	94.5±0.36	96.5±0.68	97.7±0.30	**98.0±0.47**
	κ	86.7±1.77	85.9±1.11	87.3±0.88	92.5±1.26	94.0±0.57	**94.7±1.35**
35	OA	88.8±1.21	88.0±0.59	89.1±0.56	94.0±1.52	94.6±0.72	**95.7±1.18**
	AA	94.3±0.57	93.2±0.39	94.5±0.57	96.8±0.74	97.7±0.26	**98.0±0.58**
	κ	87.6±1.33	86.6±0.65	87.8±0.62	93.3±1.69	94.0±0.79	**95.2±1.31**
40	OA	89.3±0.78	88.6±0.49	89.2±0.96	93.9±1.24	95.0±0.36	**96.4±0.79**
	AA	94.7±0.45	94.0±0.43	95.0±0.39	97.2±0.58	97.9±0.19	**98.4±0.36**
	κ	88.1±0.87	87.3±0.54	88.0±1.06	93.2±1.38	94.5±0.40	**96.0±0.88**

samples. When additional spatial information is available, the performance of the spectral-based classifiers is dramatically improved. This can be clearly seen from the class accuracies of Classes 1, 7 and 9. For these three classes with only 3 training samples in each class, the spectral classifiers obtain bad results because the spectral similarity provided by the limited training samples is insufficient to represent the whole materials. However, when the spatial neighborhood information is taken into account, the class accuracies are dramatically improved, 29.3% versus 81.6% for Class 1, 71.6% versus 96.8% for Class 7, and 52.9% versus 98.8% for Class 9 for comparing KELM and KELM-CK. The reason is that spatial information helps to discriminant the samples with similar spectral curves based on the principle: samples from the same class have similar spectral curves as well as similar spatial neighborhood structures while samples from different classes usually have different spatial neighborhood structures even if they are spectrally similar. This conclusion can also be verified on Classes 2, 3, 4, and Classes 10, 11, 12, respectively. Classes 2, 3, 4 are three subclasses of corns, i.e., "Corn-notill", "Corn-mintill", and "Corn". Classes 10, 11, 12 are three subclasses of soybeans, i.e., "Soybean-notill", "Soybean-mintill" and "Soybean-clean". These classes have much similar spectral responses so they are difficult to be separated by spectral-based classifiers. However, the spatial-spectral information helps to identify subtle critical differences, and KELM-CK achieves a good classification accuracies, i.e, more than 92% on three classes of corn, and more than 90% on three class of soybeans.

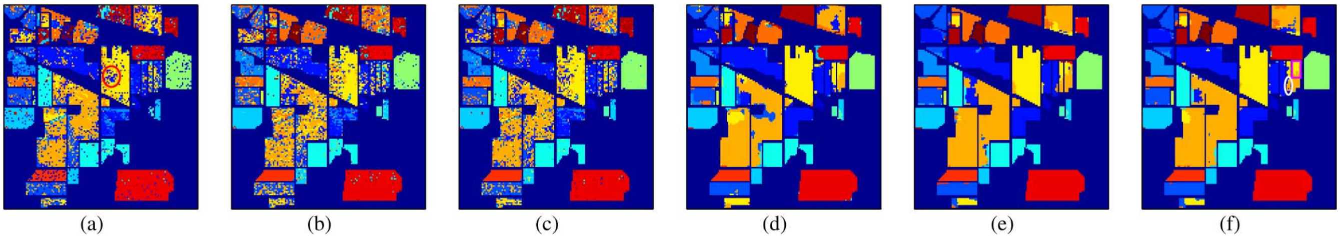


Fig. 6. Classification maps for the Indian Pines dataset with 40 labeled samples per class. (a) SVM (OA = 70.1%). (b) ELM (OA = 64.5%). (c) KELM (OA = 71.9%). (d) SVM-CK (OA = 89.1%). (e) ELM-CK (OA = 92.5%). (f) KELM-CK (OA = 93.4%).

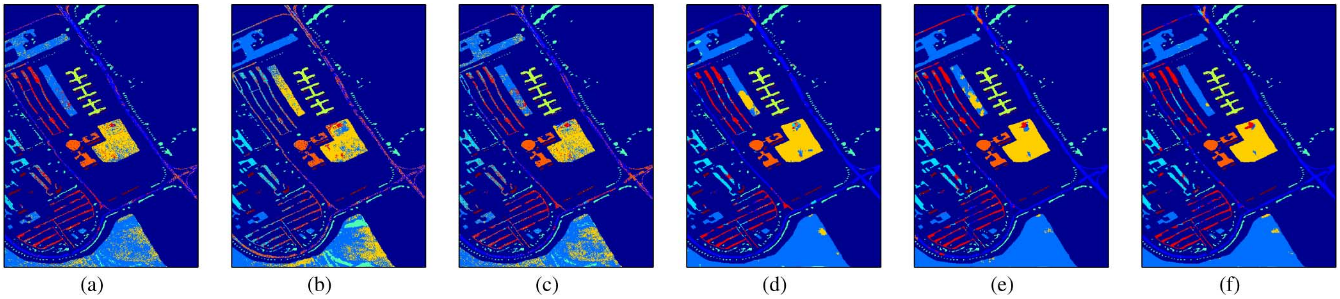


Fig. 7. Classification maps for the University of Pavia dataset with 40 labeled samples per class. (a) SVM (OA = 78.5%). (b) ELM (OA = 66.0%). (c) KELM (OA = 77.9%). (d) SVM-CK (OA = 91.4%). (e) ELM-CK (OA = 91.0%). (f) KELM-CK (OA = 93.5%).

We can also see from Table I that the proposed ELM-CK and KELM-CK methods show similar performance and outperform other classifiers consistently. Compared with SVM-CK, ELM-CK and KELM-CK increase OA by about 3%. Even for the difficult-to-separate classes of corns and soybeans, KELM-CK improves SVM-CK by more than 4% in CA. Although ELM methods don't show significant improvements over SVM on the spectral data, their counterparts on the joint spatial-spectral data have overwhelming advantages over SVM-CK. It demonstrates that ELM is more powerful to exploit the rich spatial information.

In the following, we further investigate the performance of the proposed methods under different numbers of labeled samples per class. We randomly choose $M = 5, 10, 15, 20, 25, 30, 35, 40$ samples from each class to form the training set, respectively (For the class less than M samples, half of total samples are chosen). The remaining samples form the testing set. The OA, AA, and κ values of the classification methods under different numbers of labeled samples per class for three datasets are shown in Tables II–IV, respectively. As shown in the tables, with the increase of training samples, the OA, AA and κ values for all algorithms are greatly improved. For the spectral classifiers, KELM provides better results than SVM for the Indian Pines and Salinas datasets especially in the case of extremely limited training samples, and slightly worse results than SVM on the University of Pavia dataset. ELM shows relatively bad results on three datasets. In all of the experiments, the spatial-spectral methods provide more accurate results than the spectral methods. It demonstrates that spatial information is necessary to complement the spectral features for identifying the subtle differences of similar objects. Among the spatial-spectral methods, the proposed ELM-CK and KELM-CK show a significant improvement over SVM-CK. It indicates that ELM-based methods are more powerful to exploit the rich spatial information than SVM. When the number of labeled samples per class is 5, the ELM-based CK methods improve OA of SVM-CK up to 7.8%, 6.8%, 9.1% on the Indian Pines, University of Pavia and Salinas datasets, respectively. This shows that ELM-based methods are more stable than SVM in the case of small-sized-sample.

The classification maps of different methods under 40 training samples per class for three datasets are shown in Figs. 6–8, respectively. We take the classification maps of the Indian Pines dataset as an example to visually observe the classification performance of different algorithms. It can be clearly seen that the classification maps of ELM-CK and KELM-CK are more spatially coherent in the large homogeneous regions than other methods. In addition, the spatial-spectral methods provide better results than spectral methods in terms of consistent classification results with little noise. In particular, the improvement is typically arisen for classes with similar spectral signatures. This can be seen from Fig. 6(a), where the pixels in the circled region belonging to the "Corn-mintill" class are wrongly classified to the nearby and similar class "Corn-notill" by the spectral-based SVM, while they are correctly classified by all three spatial-spectral classifiers. As we have mentioned before, the joint spatial-spectral information helps to identify subtle critical differences of spectrally similar materials. However, by implicitly assuming that the spatial neighboring pixels are similar, the spatial-spectral classifiers make mistakes on the boundary test pixels. This can be seen from Fig. 6(f), the pixels in the circled region belonging the class "Soybean-notill" (luminous yellow) are wrongly classified to the nearby class "Soybean-mintill" (ochre yellow), while the pixels in the squared region belonging the class "Soybean-mintill" are wrongly classified to the nearby class "Soybean-notill". Because a large window with width 9

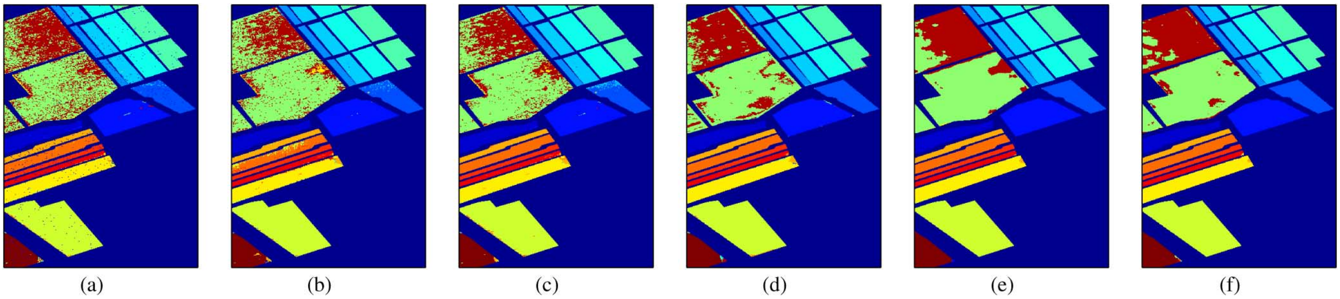


Fig. 8. Classification maps for the Salinas dataset with 40 labeled samples per class. (a) SVM (OA = 89.3%). (b) ELM (OA = 88.6%). (c) KELM (OA = 89.2%). (d) SVM-CK (OA = 93.9%). (e) ELM-CK (OA = 95.0%). (f) KELM-CK (OA = 96.4%).

is used for the spatial-spectral classifiers, the spatial neighborhood of a boundary test pixel may contain the pixels from other materials or backgrounds. Therefore, for a boundary test pixel, the local spatial feature extracted from the spatial neighborhood can not faithfully reflect its real spatial structure. If, in the spatial neighborhood of a boundary pixel, the pixels from another materials are dominant, then the spatial-spectral classifiers will make a wrong classification for the boundary pixel. If a small window is used, the boundary effect can be reduced. However, the classification performance in the large homogeneous regions will be decreased because the spatial information is insufficient to discriminant spectrally similar pixels.

V. Conclusion

In this paper, we have proposed a new ELM with the CK framework for HSI classification. In particular, ELM is performed on the joint spatial–spectral data using a linear combination of the spatial and spectral activation–function-based kernels or general Gaussian kernels. Experimental results have shown that the proposed ELM-CK and KELM-CK are more accurate and much faster than the benchmark SVM-CK for the spatial–spectral classification of HSIs.

Acknowledgment

The authors would like to thank the Editor-in-Chief, the anonymous Associate Editor, and the three reviewers for their insightful comments and suggestions that have greatly improved this paper, Prof. D. Landgrebe for providing the Indian Pines dataset, Prof. P. Gamba for providing the University of Pavia dataset, Prof. C. Lin for providing LIBSVM toolbox, and Prof. G. Huang for providing ELM codes.

References

[1] A. Plaza *et al.*, “Recent advances in techniques for hyperspectral image processing,” *Remote Sens. Environ.*, vol. 113, pp. S110–S122, 2009.

[2] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification: Earth monitoring with statistical learning methods,” *IEEE Signal Proc. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[3] G. F. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Trans. Inform. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.

[4] J. M. Bioucas-Dias *et al.*, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[5] 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. 1778–1790, Aug. 2004.

[6] G. Camps-Valls and L. Bruzzone, “Kernel-based methods for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[7] G. Mountrakis, J. Im, and G. Ogole, “Support vector machines in remote sensing: A review,” *ISPRS J. Photogramm. and Remote Sens.*, vol. 66, pp. 247–259, 2011.

[8] G. Camps-Valls, T. B. Marsheva, and D. Zhou, “Semi-supervised graph-based hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[9] B. Kuo, H. Ho, C. Li, C. Hung, and J. Taur, “A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 317–3226, Jan. 2014.

[10] Y. Gu and K. Feng, “Optimized Laplacian SVM with distance metric learning for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1109–1117, Jun. 2013.

[11] 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.

[12] J. Liu, Z. Wu, Z. Wei, L. Xiao, and L. Sun, “Spatial-spectral kernel sparse representation for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2462–2471, Dec. 2013.

[13] M. Cui, S. Prasad, W. Li, and L. M. Bruce, “Locality preserving genetic algorithms for spatial-spectral hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1688–1697, Jun. 2013.

[14] P. Quesada-Barriuso, F. Argüello, and D. B. Heras, “Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1117–1185, Apr. 2014.

[15] J. A. Benediktsson, M. Pesaresi, and K. Arnason, “Classification and feature extraction for remote sensing images from urban areas based on morphological transformations,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1940–1949, Sep. 2003.

[16] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, “Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[17] K. Tan, E. Li, Q. Du, and P. Du, “Hyperspectral image classification using band selection and morphological profiles,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 40–48, Jan. 2014.

[18] Q. Jackson and D. Landgrebe, “Adaptive bayesian contextual classification based on markov random fields,” *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2454–2463, Mar. 2002.

[19] G. Moser and S. B. Serpico, “Combining support vector machines and markov random fields in an integrated framework for contextual image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2734–2752, May 2013.

[20] G. Camps-Valls, L. Gomez-Chova, J. Muñoz Maré, J. Vila-Francés, and J. Calpe-Maravilla, “Composite kernels for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[21] M. Marconcini, G. Camps-Valls, and L. Bruzzone, “A composite semisupervised SVM for classification of hyperspectral images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 234–238, Apr. 2009.

[22] M. Fauvel, J. Chanussot, and J. A. Benediktsson, “A spatial-spectral kernel-based approach for the classification of remote-sensing images,” *Pattern Recogn.*, vol. 45, pp. 381–392, 2012.

[23] G. B. Huang, Q. Y. Zhou, and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[24] G. B. Huang, L. Chen, and C. K. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Trans. Neur. Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[25] 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.

[26] G. B. Huang, “An insight into extreme learning machines: Random neurons, random features and kernels,” *Cogn. Comput.*, vol. 6, no. 3, pp. 376–390, 2014.

[27] C. L. Philip Chen, “A rapid supervised learning neural networks for fuction interpolation and approximation,” *IEEE Trans. Neur. Netw.*, vol. 7, no. 5, pp. 1220–1230, Sep. 1996.

[28] C. L. Philip Chen and J. Z. Wan, “A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.

[29] M. Pal, “Extreme-learning-machine-based land cover classification,” *Int. J. Remote Sens.*, vol. 30, no. 14, pp. 3835–3841, Jul. 2009.

[30] L. L. C. Kasun, H. Zhou, G. B. Huang, and C. M. Vong, “Representational learning with extreme learning machine for big data,” *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Dec. 2013.

[31] J. Peng, L. Li, and Y. Tang, “Combination of activation functions in extreme learning machines for multivariate calibration,” *Chemom. Intell. Lab. Syst.*, vol. 120, pp. 53–58, Jan. 2013.

[32] H. Chen, J. Peng, Y. Zhou, L. Li, and Z. Pan, “Extreme learning machine for ranking: Generalization analysis and applications,” *Neur. Netw.*, vol. 53, pp. 119–126, May 2014.

[33] M. Pal, A. E. Maxwell, and T. A. Warner, “Kernel-based extreme learning machine for remote-sensing image classification,” *Remote Sens. Lett.*, vol. 4, no. 9, pp. 853–862, 2013.

[34] Y. Bazi *et al.*, “Differential evolution extreme learning machine for the classification of hyperspectral images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 6, pp. 1066–1070, Jun. 2014.

[35] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, “E^2LMs: Ensemble extreme learning machines for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, Apr. 2014.

[36] D. B. Heras, F. Argüello, and P. Quesada-Barriuso, “Exploring ELM-based spatial-spectral classification of hyperspectral images,” *Int. J. Remote Sens.*, vol. 35, no. 2, pp. 401–423, 2014.

[37] P. L. Bartlett, “The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network,” *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, Mar. 1998.

[38] L. Gómez-Chova, G. Camps-Valls, L. Bruzzone, and J. Calpe-Maravilla, “Mean map kernel methods for semisupervised cloud classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 207–220, Jan. 2010.

[39] D. Tuia, F. Ratle, A. Pozdnoukhov, and G. Camps-Valls, “Multisource composite kernels for urban–image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 88–92, Jan. 2010.

[40] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, “Generalized composite kernel framework for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[41] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal kernels,” *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, 2006.

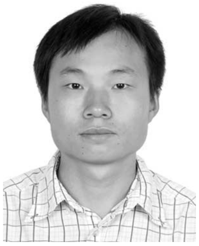
[42] S. S. Keerthi and C. J. Lin, “Asymptotic behaviors of support vector machines with Gaussian kernel,” *Neural Comput.*, vol. 15, no. 7, pp. 1667–1689, Jul. 2003.

[43] C. Chang and C. Lin, “LIBSVM : A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. Software available at: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Yicong Zhou (M’07–SM’14) received the B.S. degree from the Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees from the Tufts University, Medford, MA, USA, in 2008 and 2010, all in electrical engineering.

Currently, he is an Assistant Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include multimedia security, image/signal processing, pattern recognition, and medical imaging.

Dr. Zhou is a member of International Society for Photo-Optical Instrumentations Engineers (SPIE).



Jiangtao Peng received the B.S. degree in information and computation sciences and M.S. degree in applied mathematics from the Hubei University, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree in pattern recognition and intelligence systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011.

He is currently a Lecturer with the Faculty of Mathematics and Statistics, Hubei University. His research interests include machine learning and hyperspectral image processing.

C. L. Philip Chen (S’88–M’88–SM’94–F’07) received the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree from the Purdue University, West Lafayette, IN, USA, in 1988, both in electrical engineering.

After having worked in USA for 23 years as a Tenured Professor, the Department Head, and the Associate Dean in two different universities, Dr. Chen is currently the Dean of the Faculty of Science and Technology and a Chair Professor of the Department of Computer and Information Science with the University of Macau, Macau, China.

Dr. Chen is a Fellow of the American Association for the Advancement of Science (AAAS) and a Fellow of Hong Kong Institution of Engineers (HKIE). Currently, he is the Junior Past President of the IEEE Systems, Man, and Cybernetics Society, and the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.