Ensemble Learning for Hyperspectral Image Classification Using Tangent Collaborative Representation

Hongjun Su®, Senior Member, IEEE, Yao Yu, Qian Du®, Fellow, IEEE, and Peijun Du, Senior Member, IEEE

Abstract—Recently, collaborative representation classification (CRC) has attracted much attention for hyperspectral image analysis. In particular, tangent space CRC (TCRC) has achieved excellent performance for hyperspectral image classification in a simplified tangent space. In this article, novel Bagging-based TCRC (TCRC-bagging) and Boosting-based TCRC (TCRC-boosting) methods are proposed. The main idea of TCRC-bagging is to generate diverse TCRC classification results using the bootstrap sample method, which can enhance the accuracy and diversity of a single classifier simultaneously. For TCRC-boosting, it can provide the most informative training samples by changing their distributions dynamically for each base TCRC learner. The effectiveness of the proposed methods is validated using three real hyperspectral data sets. The experimental results show that both TCRC-bagging and TCRC-boosting outperform their single classifier counterpart. In particular, the TCRC-boosting provides superior performance compared with the TCRC-bagging.

Index Terms—Bagging, boosting, ensemble learning, hyperspectral imagery, tangent space collaborative representation.

I. INTRODUCTION

HYPERSPECTRAL remote imaging sensors are capable of capturing rich spectral information using hundreds of contiguous narrow spectral bands, providing the possibility of accurate detection and identification of ground objects [1]–[3]. Supervised classification is an intense field of research in hyperspectral data analysis [4], [5]. However, there are several challenges in hyperspectral image classification, such as limited training samples and high dimensionality [6], [7]. When training samples are insufficient, the increased number of spectral bands can frequently result in the well-known “the Hughes Phenomenon” [8]. To address these problems, one approach is to develop more advanced classifiers, for example, logical regression [9], [10], artificial neural network (ANN) [11], and extreme learning machine (ELM) [12]. Some methods such as convolutional neural network (CNN) [13] and transfer learning (TL) [14] have also been widely studied in recent years. Due to the complexity of hyperspectral images in the spectral domain, there is no guarantee that a specific classifier can accomplish the best performance in every situation.

Another alternative approach is ensemble learning that can provide enhanced classification accuracy. Ensemble learning does not refer to a specific algorithm but the concept of integrating multiple base learners to jointly determine a final result, which are deemed to be better than a base learner [15], [16]. Over the past few years, a considerable amount of literature have focused on ensemble learning in hyperspectral image classification. One of the most widely used ensemble learning approaches is Bagging, where each base classifier is trained using bootstrapped replicas of the training set. The most popular Bagging ensemble methods is random forest (RF) [17]. RF is constructed by a set of decision trees (DTs) to make a prediction. Each DT is trained on a bootstrapped sample, and features are randomly selected to split a leaf on each tree. A novel Bagging-based ELM has been proposed in [18], which exhibits the effectiveness of ensemble learning. Bagging-based RF ensemble adopts RF as a base classifier and extracts the extended multiextinction profile (E EMP) to improve performance [19]. Boosting generates a series of classifiers by reweighting training samples. ANN is used as a base weak classifier using the AdaBoost scheme in [20]. The AdaBoost-based ELM ensemble is demonstrated to be superior to the ANN ensemble owing to high capability of ELM [18]. ELM and random subspace are combined as an extension of the ELM ensemble [21]. Random subspace is an ensemble learning strategy by randomly selecting a feature set from the original one. A deep learning random subspace ensemble is developed in [22], where CNN and the deep residual network are selected as individual classifiers. The rotation-based RF ensemble (RoRF) applies data transformation and random feature selection. Boosted RoRF is constructed by integrating RoRF with the AdaBoost method, which can achieve higher accuracy [19].

In recent years, representation-based methods have shown great potential in hyperspectral image classification. Sparse representation classification (SRC) assumes that a testing sample can be sparsely represented by training samples via an
the smallest representation residual [24]. Collaborative representation applies an $l_2$-norm constraint to obtain an analytic solution, resulting in low computation cost [25], [26]. Several techniques have been applied to representation-based classifiers, which perform great capability in various applications. For example, a multimodal hypergraph learning-based sparse coding is proposed for image click prediction. Both strategies of early and late fusion of multiple features are used [27]. However, it is inappropriate to simply concatenate multiple features. High-order distance-based multiview stochastic learning combines various features into a unified representation and integrates label information into image classification [28]. Using p-Laplacian regularization to preserve the local geometry, the performance in scene recognition is greatly improved [29]. It has been proven that the manifold can greatly enhance the discrimination between different classes [30]. Tangent space CRC (TCRC), which takes advantage of the local manifold in the tangent space of the testing sample by using the simplified tangent distance, can achieve better performance than CRC [31]. However, it is sensitive to the regularization parameter.

In the field of scene processing, dictionary learning of SRC has been applied to build a basic classifier, and random subspace learning and Bagging learning algorithms are adopted [32]. The combination of multiple collaborative representation with the boosting algorithm has shown superior performance in pattern recognition [33]. A multiview boosting algorithm, called Boost.SH, computes weak classifiers independent of each view but uses a shared weight distribution to propagate information among the multiple views to ensure consistency [34]. As the most commonly used ensemble learning approach, Bagging requires that the base classifier in the learning algorithm must be unstable; the more sensitive to the training data or parameters, the higher accuracy it can achieve [35]. Therefore, it is promising to exploit TCRC as a weak base classifier in an ensemble learning system owing to its instability. Moreover, the AdaBoost algorithm can combine some base classifiers to generate a single strong classifier adaptively. TCRC can work as the base learner in the Bagging and Boosting frameworks to improve accuracy.

In this article, the mechanism of TCRC based on the Bagging approach is investigated, which involves bootstrapped sampling of the training data and diverse dictionary generation. Then a testing sample is classified by a vote of its predictions. In addition, the Boosting-based TCRC (TCRC-boosting) algorithm that embeds TCRC as a base classifier in the Boosting framework is developed; it can determine the weight of the training data and base learner dynamically via iteratively learning misclassified samples. Then a testing sample is classified by using the weighted fusion of residuals.

The main contributions of this article are as follows. 1) to the best of our knowledge, the idea of combining the collaborative representation-based classifier and ensemble methods for HSI classification is proposed for the first time and 2) the novel ensemble learning methods based on the tangent space collaborative representation classifier and Bagging/Boosting ensemble strategy are presented for HSI classification.

The remainder of this article is organized as follows. Section II introduces TCRC, Bagging and Boosting approaches. Section III proposes the Bagging-based TCRC (TCRC-bagging) and the Boosting-based TCRC (TCRC-boosting) algorithms. Section IV presents the experiment and analysis with three real hyperspectral data. Finally, Section V draws the conclusion and perspectives.

II. RELATED WORK

A. TCRC

The training data $X \in \mathbb{R}^{N \times M}$ ($N$ refers to the number of the spectral bands of HSI) contain $M$ samples that belong to $K$ classes and the corresponding dictionary $D$ is constructed using $K$ different subdictionaries as $\{D_1, D_2, \ldots, D_K\}$. The subdictionary $D_m = \{x_{mi}\}_{i=1}^{M_m}$, $m \in \{1, 2, \ldots, K\}$, and $\sum_{m=1}^{K} M_m = M$. In CRC, an approximation of a testing sample $y \in \mathbb{R}^{N \times 1}$ can be represented via the linear combination of dictionary. And $l_2$-norm is applied to constrain the collaborative coefficient $\alpha \in \mathbb{R}^{M \times 1}$. The model of CRC can be formulated as

$$\alpha = \arg \min_{\alpha} \left( \|y - D\alpha^*\|_2^2 \right) + \lambda \|\alpha^*\|_2^2.$$  \hspace{1cm} (1)

In the problem of HSI classification, it is assumed that samples from the same class reside on a low-dimensional manifold. According to this assumption, a testing sample $y$ and its possible variants are located in a low-dimensional manifold $M_l$. Thus, a transform can be denoted as

$$T(y, v) : y \in M_l \rightarrow y' \in M_l$$  \hspace{1cm} (2)

where $y \in \mathbb{R}^{N \times 1}$ and $y' \in \mathbb{R}^{N \times 1}$ represent the original and the transformed spectral features of testing sample, respectively, and $v$ reflects various variations of the spectral feature. In [30], a tangent plane is applied to approximate the local region of the manifold around $y$, and then the local manifold of the testing sample is embedded in the CRC model as

$$T(y, v) = T(y, 0) + \frac{\partial T(y, v)}{\partial v} \bigg|_{v=0} v + o(\|v\|^2) \approx y + T(y)v$$  \hspace{1cm} (3)

$$\alpha(v) = \arg \min_{\alpha, v} \left( \|y + T(y)v - D\alpha^*\|_2^2 + \lambda \|\alpha^*\|_2^2 \right)$$  \hspace{1cm} (4)

where $T(y) = (\partial T(y, v)/\partial v)|_{v=0}$ express the bases of the tangent space. As the local manifold can be approximated by the tangent plane, the manifold distance can be approximated by the tangent distance. It is suggested that the difference vector generated between a testing sample $y$ and its adjacent pixels $y' \in \{y_i | i = 1, 2, \ldots, n\}$ can be seen as the simplified tangent distance [30] as

$$\Delta y = [y'_1 - y; y'_2 - y; \ldots; y'_n - y]$$  \hspace{1cm} (5)

where $n$ is the number of adjacent pixels. If the neighboring domain is large enough and the difference vector is linearly...
independent, then
\[
\text{span}(A y) \cong \text{span}(T(y)) \quad \forall v, \exists \beta \Rightarrow T(y) v = \Delta y \beta.
\]

(7)

Su et al. [31] underlines that adding $\Delta y \beta$ to the CRC in the space can not only explore the local manifold but also increase the discrimination between different classes. In order to avoid the inversion of the singular matrix and constrain the objective point not far away from the sample point, adding a new regularizer $\|\beta\|^2$ in the objective function of the TCRC results in
\[
(\alpha, \beta) = \arg\min_{\alpha, \beta}(\|y + \Delta y \beta - D \alpha\|^2_2 + \lambda \|\alpha\|^2_2 + \eta \|\beta\|^2_2).
\]

(8)
The closed-form solution is deduced as
\[
\alpha = (D^T D + \lambda I - D^T P D)^{-1} (D^T y - D^T P y)
\]

(9)
\[
\beta = (\Delta y^T \Delta y + \eta I)^{-1} (\Delta y^T D \alpha - \Delta y^T \Delta y)
\]

(10)
where $\Delta y = \Delta y^T \Delta y$ As mentioned earlier, if the testing sample belongs to the $m$th class, the closest linear representation approximation of the testing sample is
\[
y_m = D_m (D_m^T D_m + \lambda I - D_m^T P D_m)^{-1} (D_m^T y - D_m^T P y)
\]

(11)
It yields the minimal residual error. Then the testing sample can be classified into the class yielding the minimal residual error
\[
\text{class}(y) = \arg\min_{i=1, \ldots , K} r_i(y) = \arg\min_{m=1, \ldots , K} \left(\|y + \Delta y \beta - y_m\|^2_2\right).
\]

(11)

B. Bagging

Ensemble learning is also named as the classifier ensemble or multiple classifier systems. The principle of ensemble methodology is to combine several base classifiers (each of which uses a simple learning algorithm) into a strong classifier in a certain way to obtain more accurate and reliable estimates [36]–[38]. It requires the performance of each weak classifier to be little better than random guessing. The vital component of constructing an effective ensemble learning system is producing basic classifiers with high diversity [39], [40]. Diversity requires that the generalization errors produced by the basic classifiers should be as much uncorrelated as possible. In order to approach preferred diversity, when the same classifier is selected as the base learner, three strategies are often used: manipulating the original training data, using multiple features, and increasing the diversification of algorithm parameters [41], [42].

Representative ensemble methods include Bagging, Boosting [43], and random subspace [44]–[46]. Bagging and Boosting create diversity by resampling and reweighting the original training data. Random subspace generates different training features by randomly selected subsets of the input feature space. Bagging (namely bootstrap aggregating) is one of the most intuitive and simple frameworks in ensemble learning that uses the bootstrapped sampling method. In this method, many original training samples may be repeated in the resulting training data, whereas others may be left out. Samples are selected randomly from the training set, instructive iteration is applied to create different bags, and the weak learner is trained in each bag. Each base learner predicts the label of the unknown sample, respectively. Finally, a majority voting rule is applied for the final decision. Majority voting is a simple and effective method for classifier combination. The pseudocode of the Bagging algorithm is described as follows.

Input: Train Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, $y$ represents the label of train sample, $y_i \in \{-1, +1\}$

Base learning algorithm $H$, Number of learning rounds $T$

For $t = 1, 2, \ldots T$

Generate a bootstrapped sample $D_t = \text{Bootstrap}(D)$

Train a base learner from the bootstrapped sample $h_t = H(D_t)$

End

Output: $H(x) = \arg\min \sum_{t=1}^{T} h_t(x)$

C. Boosting

Boosting is another ensemble algorithm in which the most famous one is AdaBoost. The AdaBoost algorithm can improve classification performance by iteratively learning difficult training samples. AdaBoost requires a base classifier that should be better than random guess. For this algorithm, the training step involves a serial iteration. Before iteration, a uniform weight across the training samples is initialized.

If a training sample is classified correctly by the current base classifier, then the chance of being used is reduced in a subsequent iteration, whereas the training samples classified wrongly will receive more attention from the successive base classifier. Meanwhile, the weight of the base learner with the smaller misclassification rate is larger, and those with the larger misclassification rate is smaller. After that, it generates a final strong classifier through a weighted combination of these base learners. Bagging reduces variance significantly but has little effect on bias, whereas Boosting can significantly reduce both bias and variances of the testing error. Therefore, in most cases, Boosting can produce more accurate classification results than Bagging. The pseudocode of the Boosting algorithm is described as follows.

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, $y \in \{-1, +1\}$ Base learning algorithm $H$; Number of learning rounds $T$

Initialize the weight distribution $D_1(i) = 1/m$

For $t = 1$ to $T$

Train a base learner $h_t$ from $D$ using distribution $D_t$ $h_t = H(D, D_t)$

Measure the error of $h_t$: $\epsilon_t = \text{Pr}_{x \sim D_t}[h_t(x) \neq y_i]$ Determine the weight of $h_t$: $\alpha_t = \frac{1}{T} \ln \left(1 - \epsilon_t \over \epsilon_t \right)$

Update the distribution $D_{t+1}(i)$

\[
D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} 
\exp(-\alpha_t) & \text{if} \ h_t(x_i) = y_i \\
\exp(\alpha_t) & \text{if} \ h_t(x_i) \neq y_i
\end{cases}
\]

$Z_t$ is a normalization factor which enables $D_{t+1}$ to be a distribution.

End

Output: $H(x) = \text{sign}(f(x)) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$
III. PROPOSED METHODS

A. TCRC-Bagging

In this article, the Bagging ensemble learning framework with TCRC (TCRC-bagging) as a base classifier is proposed. For the proposed TCRC-bagging algorithm, each training data of the base classifier \( \{X_1, X_2, X_3, \ldots, X_T \} \) is selected randomly from an original training set \( X \) by replacement, where \( T \) denotes the number of base classifiers (ensemble size). Owing to the bootstrap sample, a set of diverse subtraining data is generated. Then a subtraining set constructs a discrepant dictionary \( D_t = \{D_{1t}, D_{2t}, D_{3t}, \ldots, D_{Kt}\} \) in a new subdata \( X_t \). Therefore, when using the TCRC model in each subset, each bag obtains \( T \) representation coefficients \( a \) to classify the testing sample. In each bag, the closest approximation of the testing sample may be from different classes, which means that the minimal residual may be derived from different classes. The final classification result is produced by integrating the results from all bags using a majority voting rule. In this algorithm, two regularization parameters need to be predefined. The main steps of TCRC-bagging are illustrated in Algorithm 1 and its flowchart is shown in Fig. 1.

The Bagging algorithm iteratively combines weak classifiers to approximate classifiers. Diversity is obtained by using bootstrapped replicas of the training set. Making use of this diversity can effectively reduce the bias of the data. Generally, unstable classifiers are primarily used as member classifiers in the Bagging system. A little change in parameters or training samples may lead to varied predictive results. TCRC is unstable caused by the regularization parameter, so it is beneficial for TCRC to acquire better classification accuracy via the Bagging scheme.

B. TCRC-Boosting

In this section, a new Boosting framework with TCRC as a base classifier is proposed. In this algorithm, the weight of a training sample describes the contribution rate to the testing sample classification. Weights are introduced in the proposed framework to rebalance the importance of each sample per class. At each iteration of the training process, the weight of each training samples is set according to the current error rate of the classifier. In the subsequent iteration, the probabilities of training data are reduced when the weak classifiers make good predictions; otherwise, the probability of training data is increased. In this way, the AdaBoost method pays more attention to the informative or difficult training samples. Meanwhile, increasing the weight of TCRC with the larger misclassification rate are conducted adaptively. At the end of the procedure, \( T \) weighted training sets and \( T \) base classifiers are generated. According to TCRC-boosting, the testing sample \( y \) can be classified into the class that has the minimal weighted residual error

\[
\text{class}(y) = \arg \min_{i=1,2,\ldots,K} r_i(y) \quad (12)
\]

where the residual error \( r_i(y) = \sum_{t=1}^T \delta_t \|y + \Delta y\beta^t - D_t^i a^t\|_2^2 \) and \( \delta \) represents the weight of
the weak classifier TCRC. \( \hat{c} = [\hat{c}_1, \ldots, \hat{c}_T] \) is a positive vector and \( \sum_{t=1}^T \hat{c}_t = 1 \). The greater the value of \( \hat{c}_t \), the greater the weight of TCRC. Iteration is terminated if \( \hat{c}_t = 0 \) for certain \( t \). \( d_t \) measures the difference between the residual of the correct class and the minimal residual of the rest class of the training sample. Therefore, \( d_t(i) < 0 \) if the \( i \)th sample is labeled correctly. The smaller the value of \( d_t \), the better the discriminativeness of the dictionary \( X_t \). The TCRC-boosting method uses \( e_t \) to replace the error rate in the classical AdaBoost algorithm. It can better understand the degree of misclassification of each training sample, thereby adjusting the weight of each training sample more accurately. It is straightforward to see \( |\epsilon_t| < b_t \) for each iteration. The main steps of TCRC-boosting are summarized in Algorithm 2 and its flowchart is shown in Fig. 2.

IV. EXPERIMENTS AND ANALYSIS

A. Experiment Data

Three real hyperspectral data sets were used to validate the effectiveness of the proposed method. The first hyperspectral data set was obtained by the airborne Hyperspectral Mapper (HYMAP) system over a residential area near the Purdue Campus in 1999 with a spatial resolution of 5 m. The image had 377 \times 512 pixels with the wavelength ranging from 450 to 2480 nm. The scene consisted of six classes with 126 bands after removing water absorption bands [shown in Fig. 3(a) and (b)]. The second hyperspectral data set was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Northwestern Indiana. The scene, which provided 200 spectral bands with a spatial resolution of 20 m, was composed of 145 \times 145 pixels [shown in Fig. 4(a) and (b)]. It covered the 400–2500 nm range of the electromagnetic spectrum. Nine classes were used in the experiment. The last hyperspectral data set was gathered by the AVIRIS sensor over Salinas Valley, California. The image was comprised of 512 \times 217 pixels with the spatial resolution of 3.7 m. After discarding 20 water absorption bands, 204 bands were used for classification. There were 16 classes as shown in the ground truth image in Fig. 5(a) and (b).

Algorithm 2 TCRC-Boosting

Input: TCRC: base classifier, \( T \): the number of classifiers
\( X \in \mathbb{R}^{N \times M} \): Input training set and their label \( c, \lambda, \eta \): regularized parameter; \( y \): testing sample, \( \Delta y \): difference vector of \( y \), \( \Delta X \): difference vector of \( X \)

Weight initialization: \( D_t(i) = 1/M, i = 1, 2, \ldots, M \)

For \( t = 1 \) to \( T \)

Get new sample \( X' \) under distribution \( D_t \)

Construct new dictionary \( D' \) by \( X' \)

Train TCRC using new sample \( D' \) according to Eqs. (8-11)

Calculate \( \epsilon_t \) and \( b_t \):

\[ d_t = \| X + \Delta X \beta^t - D'_t \alpha'_m \|^2_2 - \min_{i \in c} \| X + \Delta X \beta^t - D'_t \alpha'_m \|^2_2 \]

\[ \epsilon_t = \text{dot}(D_t, d_t) \text{ and } b_t = \max |d_t| \]

Calculate the residual error of \( y \):

\[ r_m(y) = \| y + \Delta y \beta^t - D'_m \alpha'_m \|^2_2, m = 1, 2, \ldots K \]

Choose \( \delta_t \) as: \( \delta_t = \max(1/2b_t, 0) \)

\[ \log((b_t - \epsilon_t)/b_t + \epsilon_t), 0) \]

update the weight \( D_{t+1}(i) = (D_t(i))/(Z_t)e^{\delta_t d_t} \) where \( Z_t \) is the normalization factor

End for: The weights \( \{\delta_t\}^T_{t=1} \) after normalization

Classification: \( \text{class}(y) = \arg \min_{i=1,2,\ldots,K} \sum_{t=1}^{T} \delta_t \| y + \Delta y \beta^t - D'_t \alpha'_m \|^2_2 \)

For the three data sets, labeled samples were randomly divided for training and testing. For HYMAP data, 15 samples in each class were used as training samples. For Indian Pines data, 5% of each class was selected as training. Ten samples per class was used as training samples for Salinas data. The rest of labeled samples were used for testing.

B. Experiment Setup

In order to evaluate the performance of the two proposed methods, Bagging (base classifier is DT), Boosting (base classifier is DT), RF, CRC, and TCRC are chosen for
the comparison purpose in the experiments. The Bagging, Boosting, and RF classifiers are operated for 20 iterations. Seven methods are repeatedly operated ten times and the averaged result is reported.

### C. Classification Performance

The optimal parameters for the proposed methods are listed in Table I. $T$ refers to the size of ensemble (i.e., the number of classifiers). The number of $T$ is fixed to 20 in the experiments. The value of $n$ denotes the number of neighboring pixels. According to [31], the number of $n$ is set as 8. The classification accuracies including OA, AA, Kappa statistic, and individual class accuracy are reported in Tables II–IV. The best results are shown in bold.

For the HYMAP urban data, Fig. 3(c)–(i) visually displays the seven classification maps. From Table II, the OA (%) values for Bagging, Boosting, RF, CRC, TCRC, TCRC-bagging, and TCRC-boosting are 87.77, 86.09, 87.31, 91.48, 90.91, 92.59, and 93.73, respectively. It is clear that the two proposed TCRC-bagging and TCRC-boosting algorithms outperform their single classifier counterpart because

---

**Fig. 3.** Purdue campus. (a) False-color image. (b) Ground truth. Classification maps obtained by (c) Bagging, (d) Boosting, (e) RF, (f) CRC, (g) TCRC, (h) TCRC-bagging, and (i) TCRC-boosting.

**TABLE I**

<table>
<thead>
<tr>
<th>Data</th>
<th>Purdue Campus</th>
<th>Indian Pines</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
<td>$\lambda$ 1e-5</td>
<td>1e-4</td>
<td>1e-4</td>
</tr>
<tr>
<td>TCRC</td>
<td>$\lambda$ 1e-6</td>
<td>1e-5</td>
<td>1e-5</td>
</tr>
<tr>
<td></td>
<td>$\eta$ 1e-8</td>
<td>1e-8</td>
<td>1e-8</td>
</tr>
<tr>
<td></td>
<td>$n$ 8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>TCRC-bagging</td>
<td>$\lambda$ 1e-7</td>
<td>1e-5</td>
<td>1e-9</td>
</tr>
<tr>
<td></td>
<td>$\eta$ 1e-8</td>
<td>1e-8</td>
<td>1e-8</td>
</tr>
<tr>
<td></td>
<td>$n$ 8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>$T$ 20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>TCRC-boosting</td>
<td>$\lambda$ 1e-9</td>
<td>1e-9</td>
<td>1e-8</td>
</tr>
<tr>
<td></td>
<td>$\eta$ 1e-8</td>
<td>1e-8</td>
<td>1e-8</td>
</tr>
<tr>
<td></td>
<td>$n$ 8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>$T$ 20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
using ensemble learning we can obtain more accurate and reliable classification accuracy. The OA value of TCRC-boosting obtains the highest OA, AA, and Kappa statistic, with the improvement of 2.97%, 2.01%, and 0.032, respectively, over the TCRC method. The OA value of TCRC-bagging obtains the second highest OA, AA, and Kappa, with the improvement of 2.18%, 1.2%, and 0.0179, respectively, over the TCRC method. Additionally, TCRC-boosting classifiers provide the best results in terms of overall and individual class accuracy, followed by TCRC-bagging and TCRCRC. The experiment results indicate that Boosting algorithms outperform Bagging algorithms when the base learner is TCRC.

For the second data, the seven classification maps are shown in Fig. 4(c)–(i). Table III reports the detailed classification results of the proposed methods and other classifiers. The OA (%) values for seven classifiers Bagging, Boosting, RF, CRC, TCRC, TCRC-bagging, and TCRC-boosting are 71.72, 69.48, 71.05, 68.96, 80.14, 81.14, and 84.11, respectively. Compared with Bagging, Boosting and RF methods, the TCRC and TCRC ensembles offer dramatic improvement for the Indian Pines data set. Similar to the HYMAP data set, TCRC-boosting can also provide the best performance by producing 84% overall classification accuracy. It is worth noting that the overall accuracy of TCRC-boosting increases 4% over the TCRC. Meanwhile, the overall accuracy of TCRC-bagging increases 1% over TCRC. The performance gain demonstrates the effectiveness of combining several base classifiers. The TCRC-boosting also outperforms TCRC-bagging in the experiment.

For the Salinas data, the classification maps of different methods are shown in Fig. 5(c)–(i). Table IV reports the detailed classification results of the proposed methods and other classifiers. The OA (%) values for seven classifiers Bagging, Boosting, RF, CRC, TCRC, TCRC-bagging, and TCRC-boosting are 80.28, 77.16, 80.11, 81.22, 84.53, 85.32, and 86.34, respectively. Similar to Purdue Campus and Indian Pines data set, TCRC-bagging and TCRC-boosting can outperform other comparative methods, yielding about 85% and 86% classification accuracies, respectively. There are nearly 1% and
Fig. 5. Salinas (a) false-color image and (b) ground truth. Classification maps obtained by (c) Bagging, (d) Boosting, (e) RF, (f) CRC, (g) TCRC, (h) TCRC-bagging, and (i) TCRC-boosting.

2% improvements from our methods compared with the original TCRC, demonstrating the effectiveness of the ensemble learning. It can be seen that the accuracy of TCRC-boosting is higher than TCRC-bagging, especially for classes 8, 9, 10, and 13.

**D. Computing Time**

To further compare the complexity, the computing times when the algorithms run in a personal computer with 2.6-GHz CPU and 8.0-GB memory are recorded in Tables II–IV. All the experiments were carried out using the MATLAB software. We can see that the two proposed methods cost much more computing time as expected. TCRC-boosting consumes more time than TCRC-bagging as base classifiers are generated in the Boosting framework. The RF algorithm has the fastest speed, followed by the Bagging and Boosting algorithms.

**E. Parameter Analysis**

The impact of different ensemble times $T$ on the classification performance is investigated. The ensemble times were set as $T = 10, 20, 40, \text{ and } 60$ for the three data sets in the experiment. Other parameters are fixed as shown in Table IV. The zero value of the $x$ coordinate represents the original single classifier counters (TCRC). From Fig. 6(a) to (c), we can see that the overall accuracy was obviously improved by increasing the number of base classifiers between 10 and 20 for all the three data sets. No further significant increase takes place for a larger ensemble for both TCRC-bagging and TCRC-boosting. With the increase of ensemble times,
the number of base classifiers increases, but the diversity between base classifiers also decreases, resulting in poor ensemble performance. Considering the ensemble performance and time consumption comprehensively, the number of $T$ is adopted as 20 in Tables II–IV. From Fig. 6(a), it can be seen that the accuracy of TCRC-boosting increases, with the number of $T$ becoming larger. The classification accuracy of TCRC-boosting reaches about 94% when $T = 60$. It is noticed that the accuracy of TCRC-boosting becomes lower when the value of $T$ is greater than 20 in the Indian Pines data set and Salinas data set. As shown in Fig. 6(b), the overall accuracy of TCRC-boosting even drops to 81.5% sharply. The possible reason is that the Indian Pines data set is more complicated than others; it may contain high within-class variation, and then the TCRC-boosting algorithm overfits the training data.

Regularization parameters also have affected the accuracy for representation-based methods significantly. The sensitivity of the two proposed methods with varying regularization parameter $\lambda$ is evaluated for all the three data sets. In the experiments, $\lambda$ is set in the range of $1e^{-9}$–$1e^{-4}$. Other parameters have been fixed as shown in Table I. Fig. 7(a)–(c) illustrates the overall classification accuracy tendency of TCRC-bagging and TCRC-boosting with the different parameter $\lambda$ for all the three data sets. It is noticed that the OA values from the TCRC-bagging increase with the growing regularization parameter and then gradually decrease after reaching the maximum. Note that TCRC-bagging performs better in the large range ($1e^{-8}$–$1e^{-4}$); nevertheless, TCRC-boosting provides better performance in the small range ($1e^{-9}$–$1e^{-8}$). Obviously, TCRC-boosting is more sensitive to $\lambda$ than TCRC and TCRC-bagging. When $\lambda = 1e^{-4}$, the OA of TCRC-boosting even drops to 17%. In addition, it can be seen that the most suitable $\lambda$ set in TCRC-bagging for the three data sets is $1e^{-7}$, $1e^{-5}$, and $1e^{-9}$, respectively. TCRC-boosting offers the best performance for the three data sets when $\lambda = 1e^{-9}$, $\lambda = 1e^{-8}$, and $\lambda = 1e^{-8}$, respectively.

Fig. 8 analyzes the overall accuracy versus varying regularization parameter $\eta$ under the optimal parameter $\lambda$ and $n$. In the experiment, $\eta$ is in the range of $1e^{-8}$–1. Other parameters are fixed as shown in Table I. Fig. 8(a)–(c) illustrates the sensitiveness of the TCRC-bagging and TCRC-boosting to parameter $\eta$ in all the three data sets. We can see that TCRC-boosting performs more robust than TCRC-bagging and TCRC for all the three data sets. For the proposed TCRC-bagging, the most suitable $\eta$ for the three data sets.

### Table II

<table>
<thead>
<tr>
<th>Class</th>
<th>Train</th>
<th>Test</th>
<th>Bagging</th>
<th>Boosting</th>
<th>RF</th>
<th>CRC</th>
<th>TCRC</th>
<th>TCRC-bagging</th>
<th>TCRC-boosting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road</td>
<td>15</td>
<td>1272</td>
<td>85.60</td>
<td>82.60</td>
<td>82.20</td>
<td>88.15</td>
<td>88.91</td>
<td>90.70</td>
<td>92.44</td>
</tr>
<tr>
<td>Grass</td>
<td>15</td>
<td>1099</td>
<td>91.90</td>
<td>89.78</td>
<td>90.93</td>
<td>91.77</td>
<td>85.54</td>
<td>87.26</td>
<td>89.49</td>
</tr>
<tr>
<td>Shadow</td>
<td>15</td>
<td>204</td>
<td>95.39</td>
<td>71.03</td>
<td>94.02</td>
<td>93.53</td>
<td>90.46</td>
<td>90.72</td>
<td>88.43</td>
</tr>
<tr>
<td>Soil</td>
<td>15</td>
<td>364</td>
<td>93.52</td>
<td>73.24</td>
<td>94.45</td>
<td>89.70</td>
<td>86.82</td>
<td>87.98</td>
<td>91.19</td>
</tr>
<tr>
<td>Tree</td>
<td>15</td>
<td>1336</td>
<td>93.62</td>
<td>98.68</td>
<td>93.49</td>
<td>97.31</td>
<td>98.18</td>
<td>98.65</td>
<td>98.89</td>
</tr>
<tr>
<td>Roof</td>
<td>15</td>
<td>1270</td>
<td>77.32</td>
<td>83.72</td>
<td>79.67</td>
<td>90.88</td>
<td>94.63</td>
<td>96.45</td>
<td>96.17</td>
</tr>
<tr>
<td>Overall Accuracy</td>
<td>87.77</td>
<td>86.09</td>
<td>87.31</td>
<td>91.48</td>
<td>90.91</td>
<td>92.59</td>
<td>93.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Accuracy</td>
<td>89.56</td>
<td>83.18</td>
<td>89.13</td>
<td>91.89</td>
<td>90.76</td>
<td>91.96</td>
<td>92.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kappa</td>
<td>0.8466</td>
<td>0.8257</td>
<td>0.8413</td>
<td>0.8942</td>
<td>0.8885</td>
<td>0.9064</td>
<td>0.9208</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>0.51</td>
<td>1.25</td>
<td>0.10</td>
<td>2.73</td>
<td>3.07</td>
<td>121.10</td>
<td>126.95</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Class</th>
<th>Train</th>
<th>Test</th>
<th>Bagging</th>
<th>Boosting</th>
<th>RF</th>
<th>CRC</th>
<th>TCRC</th>
<th>TCRC-bagging</th>
<th>TCRC-boosting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>72</td>
<td>1356</td>
<td>61.43</td>
<td>55.41</td>
<td>57.69</td>
<td>64.69</td>
<td>77.25</td>
<td>77.10</td>
<td>84.85</td>
</tr>
<tr>
<td>C2</td>
<td>42</td>
<td>788</td>
<td>46.85</td>
<td>62.28</td>
<td>45.65</td>
<td>65.60</td>
<td>82.03</td>
<td>80.67</td>
<td>75.27</td>
</tr>
<tr>
<td>C3</td>
<td>25</td>
<td>458</td>
<td>74.62</td>
<td>73.51</td>
<td>73.44</td>
<td>91.85</td>
<td>97.59</td>
<td>97.41</td>
<td>95.83</td>
</tr>
<tr>
<td>C4</td>
<td>37</td>
<td>693</td>
<td>91.05</td>
<td>89.22</td>
<td>91.23</td>
<td>90.97</td>
<td>94.47</td>
<td>95.43</td>
<td>97.29</td>
</tr>
<tr>
<td>C5</td>
<td>24</td>
<td>454</td>
<td>95.28</td>
<td>94.05</td>
<td>97.05</td>
<td>97.79</td>
<td>99.52</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C6</td>
<td>49</td>
<td>923</td>
<td>56.16</td>
<td>58.46</td>
<td>51.53</td>
<td>64.20</td>
<td>80.01</td>
<td>80.10</td>
<td>73.18</td>
</tr>
<tr>
<td>C7</td>
<td>123</td>
<td>2332</td>
<td>77.68</td>
<td>63.06</td>
<td>79.60</td>
<td>53.19</td>
<td>66.03</td>
<td>68.11</td>
<td>75.16</td>
</tr>
<tr>
<td>C8</td>
<td>30</td>
<td>563</td>
<td>31.79</td>
<td>52.91</td>
<td>35.99</td>
<td>54.73</td>
<td>81.43</td>
<td>85.52</td>
<td>84.19</td>
</tr>
<tr>
<td>C9</td>
<td>64</td>
<td>1201</td>
<td>96.72</td>
<td>94.81</td>
<td>95.27</td>
<td>96.44</td>
<td>98.49</td>
<td>98.73</td>
<td>99.14</td>
</tr>
<tr>
<td>Overall Accuracy</td>
<td>71.72</td>
<td>69.48</td>
<td>71.05</td>
<td>68.96</td>
<td>80.14</td>
<td>81.14</td>
<td>84.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Accuracy</td>
<td>70.23</td>
<td>71.53</td>
<td>69.72</td>
<td>75.50</td>
<td>86.31</td>
<td>87.01</td>
<td>87.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kappa</td>
<td>0.6639</td>
<td>0.6371</td>
<td>0.6553</td>
<td>0.6224</td>
<td>0.7628</td>
<td>0.7754</td>
<td>0.8120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>1.41</td>
<td>1.22</td>
<td>0.35</td>
<td>3.16</td>
<td>16.11</td>
<td>3423.53</td>
<td>3651.34</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The impact of the number of training samples per class on the classification performance is also investigated. The sizes of training samples per class are set as \{5; 10; 15; 20\}. Table V reports the overall classification accuracy tendency with different training sample sizes in the three data sets. It can be seen that the accuracy of both proposed methods sets are 1e-6, 1e-8, and 1e-8, respectively. The TCRC-boosting performs the best when \( \eta = 1e-8 \) for the three data sets.

**TABLE IV**

<table>
<thead>
<tr>
<th>Class</th>
<th>Train</th>
<th>Test</th>
<th>Bagging</th>
<th>Boosting</th>
<th>RF</th>
<th>CRC</th>
<th>TCRC</th>
<th>TCRC-bagging</th>
<th>TCRC-boosting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>1999</td>
<td>97.02</td>
<td>84.59</td>
<td>96.73</td>
<td>98.28</td>
<td>99.91</td>
<td>99.92</td>
<td>99.97</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3716</td>
<td>96.85</td>
<td>97.05</td>
<td>97.87</td>
<td>98.93</td>
<td>99.28</td>
<td>99.47</td>
<td>99.29</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1966</td>
<td>87.72</td>
<td>70.90</td>
<td>87.12</td>
<td>93.75</td>
<td>97.73</td>
<td>95.57</td>
<td>97.74</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>1384</td>
<td>98.44</td>
<td>89.16</td>
<td><strong>98.15</strong></td>
<td>92.90</td>
<td>98.10</td>
<td>97.66</td>
<td>97.95</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2668</td>
<td>92.28</td>
<td>91.03</td>
<td>92.31</td>
<td>85.98</td>
<td>94.49</td>
<td>95.02</td>
<td><strong>95.52</strong></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>3949</td>
<td>98.74</td>
<td>99.25</td>
<td>98.28</td>
<td>99.58</td>
<td>99.89</td>
<td>99.97</td>
<td><strong>100</strong></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>3589</td>
<td>98.28</td>
<td>85.50</td>
<td>97.26</td>
<td>99.46</td>
<td>99.73</td>
<td>99.82</td>
<td><strong>99.97</strong></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>11261</td>
<td>60.28</td>
<td>61.71</td>
<td>58.29</td>
<td>65.89</td>
<td>66.98</td>
<td>70.83</td>
<td>73.03</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6193</td>
<td>96.43</td>
<td>96.74</td>
<td>94.35</td>
<td>91.58</td>
<td>94.92</td>
<td>95.20</td>
<td><strong>98.37</strong></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3268</td>
<td>75.09</td>
<td>73.81</td>
<td>75.04</td>
<td>82.88</td>
<td>90.38</td>
<td>92.43</td>
<td><strong>95.91</strong></td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>1085</td>
<td>81.10</td>
<td>70.64</td>
<td>83.63</td>
<td>90.52</td>
<td>95.19</td>
<td>95.41</td>
<td><strong>96.25</strong></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>1917</td>
<td>95.76</td>
<td>90.46</td>
<td>96.76</td>
<td>93.01</td>
<td>97.14</td>
<td>95.09</td>
<td>93.76</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>906</td>
<td>96.71</td>
<td>85.12</td>
<td>95.30</td>
<td>68.22</td>
<td>79.28</td>
<td>78.90</td>
<td>92.60</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>1060</td>
<td>90.58</td>
<td>83.81</td>
<td><strong>91.68</strong></td>
<td>73.79</td>
<td>83.56</td>
<td>85.17</td>
<td>90.13</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>7258</td>
<td>48.31</td>
<td>46.79</td>
<td>52.05</td>
<td>52.50</td>
<td><strong>59.76</strong></td>
<td>58.57</td>
<td>57.12</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>1797</td>
<td>88.16</td>
<td>83.05</td>
<td>87.17</td>
<td>97.74</td>
<td><strong>100</strong></td>
<td>100</td>
<td>99.98</td>
</tr>
</tbody>
</table>

**Overall Accuracy**

| Average Accuracy | 80.28 | 77.16 | 80.11 | 81.22 | 84.53 | 85.32 | **86.34** |
| kappa            | 0.7810| 0.7466| 0.7795| 0.7943| 0.8323| 0.8414| **0.8515** |
| time             | 4.08  | 6.48  | 0.33  | 55.03 | 71.91 | 289.14| 297.03    |

Fig. 6. Overall accuracy of different algorithms for two data sets with varying \( T \). (a) Purdue campus. (b) Indian Pines. (c) Salinas.

Fig. 7. Overall accuracy of different algorithms for two data sets with varying \( \lambda \). (a) Purdue campus. (b) Indian Pines. (c) Salinas.
TABLE V

<table>
<thead>
<tr>
<th>Data</th>
<th>Purdue Campus</th>
<th>Indian Pines</th>
<th>Salinas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Per class</td>
<td>TCRC</td>
<td>TCRC-bagging</td>
<td>TCRC-boosting</td>
</tr>
<tr>
<td>5</td>
<td>81.91</td>
<td>82.90</td>
<td>83.71</td>
</tr>
<tr>
<td>10</td>
<td>86.70</td>
<td>88.99</td>
<td>89.45</td>
</tr>
<tr>
<td>15</td>
<td>90.91</td>
<td>92.59</td>
<td>93.73</td>
</tr>
<tr>
<td>20</td>
<td>91.17</td>
<td>92.50</td>
<td>93.20</td>
</tr>
</tbody>
</table>

increases with the sizes of training samples. The proposed TCRC ensemble outperforms TCRC in terms of higher OA in all cases. TCRC-boosting performs better than TCRC-bagging and TCRC in Purdue Campus and Salinas data set. It is noticed that TCRC-bagging yields better classification performance than TCRC-boosting for the Indian Pines data set owing to the fact that the Indian Pines data set may contain high within-class variation, and then the TCRC-boosting algorithm overfits the training data.

V. CONCLUSION

In this article, the tangent space collaborative representation-based ensemble learning methods for hyperspectral image classification, that is, TCRC-bagging and TCRC-boosting, are proposed. For the proposed methods, TCRC is used as a weak learner to make predictions and then it is combined to make a strong classifier. Although ensemble learning requires more training time than individual classifiers, its performance is superior to individual classifiers. The experiment results show that TCRC-bagging and TCRC-boosting can achieve better performance than the state-of-the-art classifiers. Furthermore, the Boosting framework is more effective than the Bagging framework in most cases. It is confirmed that ensemble learning is an excellent option for hyperspectral image classification.

ACKNOWLEDGMENT

The authors would like to thank Professor D. Landgrebe for providing the AVIRIS data.

REFERENCES


Fig. 8. Overall accuracy of different algorithms for two data sets with varying $\eta$. (a) Purdue campus. (b) Indian Pines. (c) Salinas.


Hongjun Su (S’09–M’11–SM’18) received the B.S. degree in geography information system from the China University of Mining and Technology, Xuzhou, China, and the Ph.D. degree in cartography and geographic information system from the Key Laboratory of Virtual Geographic Environment, (Ministry of Education), Nanjing Normal University, Nanjing, China, in 2006 and 2011, respectively. He was a Visiting Student with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA, from 2009 to 2010. He was also a Post-Doctoral Fellow with the Department of Geographical Information Science, Nanjing University, Nanjing, from 2012 to 2014. He is currently an Associate Professor with the School of Earth Sciences and Engineering, Hohai University, Nanjing. He is also a Research Scholar with the Department of Geography, University of Wisconsin–Madison, Madison, WI, USA, from 2019 to 2020. He has authored more than 60 journal articles. His main research interests include hyperspectral remote sensing dimensionality reduction, classification, and spectral unmixing.

Dr. Su is an Associate Editor of *IEEE ACCESS*, a Guest Editor of the *Remote Sensing* and the *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*. He received the 2016 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. He is an Active Reviewer of the *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, the *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, the *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, and the *International Journal of Remote Sensing*.

Yao Yu received the B.E. degree from the College of Science, Central South University of Forestry and Technology, Changsha, China, in 2017. She is currently pursuing the M.E. degree in photogrammetry and remote sensing with the School of Earth Sciences and Engineering, Hohai University, Nanjing, China.

Her research interests include hyperspectral remote sensing imagery, collaborative representation, and machine learning in hyperspectral remote sensing imagery classification.
Qian Du (S'98–M'00–SM'05—F'18) received the Ph.D. degree in electrical engineering from the University of Maryland at Baltimore, Baltimore, MD, USA, in 2000.

She is currently the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. She is also an Adjunct Professor with the College of Surveying and Geo-Informatics, Tongji University, Shanghai, China. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du is a fellow of the IEEE-Institute of Electrical and Electronics Engineers, and a fellow of the SPIE-International Society for Optics and Photonics. She received the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. She was the Co-Chair of the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society from 2009 to 2013, and the Chair of the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She has served as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the Journal of Applied Remote Sensing, and the IEEE SIGNAL PROCESSING LETTERS. Since 2016, she has been the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She is also the General Chair of the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Shanghai, in 2012.

Peijun Du (M’07–SM’12) received the Ph.D. degree in geodesy and survey engineering from the China University of Mining and Technology, Xuzhou, China, in 2001.

He is currently a Professor of photogrammetry and remote sensing with the Department of Geographic Information Sciences, Nanjing University, Nanjing, China, and also the Deputy Director of the Key Laboratory for Satellite Surveying Technology and Applications, National Administration of Surveying and Geoinformation, China. He has authored more than 70 articles in international peer-reviewed journals, and more than 100 articles in international conferences and Chinese journals. His research interests include remote sensing image processing and pattern recognition, hyperspectral remote sensing, and applications of geospatial information technologies.