

Research Article

Combined Kernel-Based BDT-SMO Classification of Hyperspectral Fused Images

Fenghua Huang^{1,2} and Luming Yan²

¹ Sunshine College, Fuzhou University, Fuzhou, Fujian 350015, China

² College of Geographical Sciences, Fujian Normal University, Fuzhou, Fujian 350007, China

Correspondence should be addressed to Fenghua Huang; fenghuait@sina.com

Received 13 June 2014; Revised 28 July 2014; Accepted 29 July 2014; Published 27 August 2014

Academic Editor: Javier Plaza

Copyright © 2014 F. Huang and L. Yan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the poor generalization and flexibility problems that single kernel SVM classifiers have while classifying combined spectral and spatial features, this paper proposed a solution to improve the classification accuracy and efficiency of hyperspectral fused images: (1) different radial basis kernel functions (RBFs) are employed for spectral and textural features, and a new combined radial basis kernel function (CRBF) is proposed by combining them in a weighted manner; (2) the binary decision tree-based multiclass SMO (BDT-SMO) is used in the classification of hyperspectral fused images; (3) experiments are carried out, where the single radial basis function- (SRBF-) based BDT-SMO classifier and the CRBF-based BDT-SMO classifier are used, respectively, to classify the land usages of hyperspectral fused images, and genetic algorithms (GA) are used to optimize the kernel parameters of the classifiers. The results show that, compared with SRBF, CRBF-based BDT-SMO classifiers display greater classification accuracy and efficiency.

1. Introduction

Hyperspectral remote sensing images have great spectral resolution and ground-object recognition capabilities because they have hundreds of fine and continuous wave bands. However, these images suffer from some problems, for example, the low spatial resolution, the high proportion mixed pixels, the huge amount of data, the uncertainty of data, and the Hughes phenomenon affecting the supervised classification. Hence, it is ineffective to classify based on spectral features alone. To improve the spatial resolution, hyperspectral images can be fused with high spatial resolution images, and the classification accuracy can be improved by combining spectral features with spatial features. The support vector machine is a well-established learning method that provides an approach to the traditional Hughes effect and overfitting problems. SVM has evident advantages and thus is widely used for classifying hyperspectral remote sensing images, especially for small samples, nonlinear, and high-dimensional pattern recognition problems. However, the traditional kernel function is a single kernel function.

Single kernel-based SVM classifiers deliver lackluster performance in classifying combined spectral and spatial features. Multi-kernel SVM classifiers are more effective than single kernel classifiers because the former can combine kernel functions according to different classification features and improve classification performance by combining the advantages of spectral, spatial, and structural features. In this paper, a binary decision tree-based multiclass SMO (BDT-SMO) classifier is proposed to improve the classification accuracy and efficiency of hyperspectral remote sensing images. Two radial basis functions (RBFs) are employed for spectral and textural features and are then combined in a weighted manner to form a new combined radial basis function (CRBF). Eventually, the single radial basis function- (SRBF-) based BDT-SMO classifier and the CRBF-based BDT-SMO classifier are used, respectively, to classify hyperspectral fused images of land usages, and genetic algorithms are used to optimize the kernel parameters of classifiers. The results show that CRBF-based BDT-SMO classifiers have superior classification accuracy and efficiency.

2. Support Vector Machines (SVM)

SVM is a statistical learning-based machine learning method. The key idea behind an SVM is to take the Vapnik structural risk minimization (SRM) as the inductive principle [1]. If the feature space is linearly separable, the optimal classification hyperplane can be constructed in the high-dimensional feature space as the decision plane to maximize the distance between the two classes of data. Otherwise, kernel functions can be used to map the data into the high-dimensional feature space and then construct the optimal classification hyperplane in the high-dimensional feature space [1]. The optimal classification hyperplane has to be able to separate the two classes (the error rate is 0 during the training) and maximize the distance between the two classes [2]. An example of the SVM optimal classification hyperplane is shown in Figure 1 [3].

In Figure 1, circles and squares represent the two respective classes of samples. Let H be the optimal classification line. The points in the lines H_1 and H_2 parallel to H are support vectors, and the margin denotes the classification distance. Consider that for the classification plane $w \cdot x + b = 0$, the following is true: $y_i[(w \cdot x_i) + b] - 1 \geq 0, i = 1, \dots, n$. In this case, the classification distance is $2/\|w\|$. Maximizing the distance is equal to minimizing $\|w\|^2$. The classification plane that satisfies this condition is the optimal classification plane.

For the linearly separable sample set $(x_i, y_i), i = 1, 2, \dots, n, x \in R^d, y \in \{-1, +1\}$, the Lagrange optimization method can be used to convert the classification plane optimization problem into a dual convex quadratic optimization problem. Hence, the linear sample classification problem is equal to the maximization problem of the following function [2]:

$$Q(a) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j (x_i \cdot x_j), \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^n y_i a_i = 0, \quad a_i \geq 0, \quad i = 1, \dots, n, \quad (2)$$

where a_i denotes the Lagrange multiplier of each sample. This is a quadratic function optimization problem subject to the inequality constraints. In the solutions to (1), only a small subset of i is not equal to 0, and the corresponding samples are the support vectors. The classification plane is determined by the support vectors. That is, a small number of samples suffice to construct the optimal classification plane. Solving the above problem results in the following optimal classification function [3]:

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^n a_i^* y_i (x_i \cdot x) + b^* \right\}, \quad (3)$$

where a_i^* denotes the nonzero samples (i.e., support vectors) and b^* is the classification threshold that can be computed using any support vector or is defined as the median of a pair of support vectors from both classes.

For nonlinear classification, nonlinear mapping (i.e., the kernel function) can be performed to map the input data

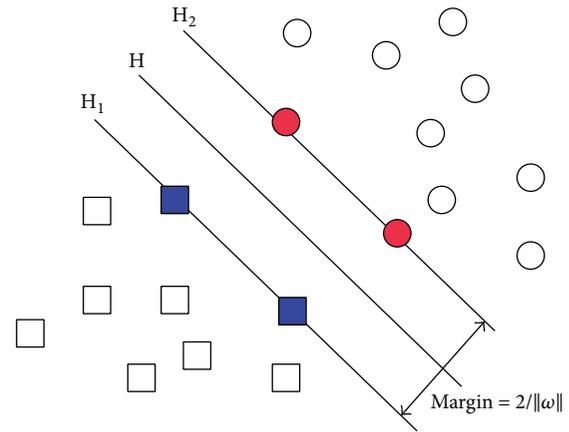


FIGURE 1: An example of the SVM optimal classification hyperplane.

from the original space, R^n , to the high-dimensional feature space, Ω , and then construct the hyperplane in the high-dimensional feature space [1]. In this case, a kernel function $K(x_i, x_j)$ must be constructed in the original space and be made equal to the inner product of the transformed high-dimensional feature space. In this instance, the nonlinear classification problem is converted to the maximization of the following function [4]:

$$Q(a) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j K(x_i, x_j), \quad (4)$$

$$\text{s.t.} \quad \sum_{i=1}^n y_i a_i = 0, \quad 0 \leq a_i \leq C \quad (i = 1, \dots, n),$$

where C is a constant. Solving the above problem will result in the following classification criterion function [5]:

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^n a_i^* y_i K(x_i, x) + b^* \right\}. \quad (5)$$

For nonlinear classification, choosing different inner kernel functions will produce different SVMs and have a different optimal classification hyperplane in the feature space. It has been shown that the common kernels are suitable for most nonlinear classification problems. The four common kernel functions are the linear kernel, the polynomial kernel, the radial basis function (RBF), and the sigmoid function [6]. The choice of SVM kernels and setup of the kernel parameters are largely dependent on empirical and experimental analysis, because no well-established methods are currently available for this. Foody and Mathur showed that the kernel parameters and the error penalty factor, C , rather than the class of the chosen kernels are the decisive factors in the performance of SVM [6].

Because traditional SVM algorithm is only suitable for two-class classification, the multiclass classification is often completed through the specific combination of multiple SVM classifiers. The multiclass classification methods commonly used are "one against one" SVM (OAO-SVM), "one against

all” SVM (OAA-SVM), error correction code SVM (ECC-SVM), directed acyclic graph SVM (DAG-SVM), and binary decision tree SVM (BDT-SVM) [7].

3. Sequential Minimal Optimization (SMO)

Traditional SVM algorithms can be boiled down to quadratic programming problems. Most existing approaches to the training of large-scale sample sets are based on decomposition and iteration, that is, solving the original quadratic programming problem by decomposing it into smaller quadratic programming problems. This method enormously increases computation complexity without guaranteeing accuracy [8]. The most prominent feature of SMO algorithms is that there is no need to run iterated algorithms because each small optimization problem can be solved analytically. Although the solution to each small optimization problem is not necessarily the final solution of the optimized Lagrange multiplier, the objective function approaches the minimal value [9]. The Lagrange multiplier can be minimized repeatedly until the KKT conditions are satisfied and the objective function is minimized. The procedures of the SMO training algorithm are shown in Figure 2 [8]. From Figure 2, it can be seen that SMO training involves firstly the determination of the two Lagrange multipliers a_1 and a_2 and the computation of a_2 's upper bound, H , and lower bound, L . Secondly, a_2 needs to be updated. If Δa_2 is smaller than the threshold, the updating fails. Otherwise, a_1 needs to be updated according to relevant rules, and the values of all objective functions, F_i , should also be updated. The deviation between the function's output and the target classification, E , will be computed. If E is smaller than the threshold, the algorithm terminates. SMO algorithms have been designed to solve two problems: firstly, they can solve simple optimization problems using analytical methods and secondly, they can choose strategies for optimizing the Lagrange multipliers [10]. Although SMO has more quadratic programming subproblems, it adopts the “divide and conquer” strategy. Its learning can therefore be performed in a parallel manner, enabling it to handle heavy matrix operations (e.g., inner product computation of the kernel functions) and sample searches. Hence, its processing speed is improved substantially. SMO algorithms can solve large-scale SVM training problems because they do not need to process large matrices and have no special requirements on memory space [10]. The SMO algorithm is also only suitable for two-class classification like traditional SVM algorithm, the multiclass classification is often completed through the specific combination of multiple SMO classifiers, and the multiclass classification methods commonly used are OAO-SMO, OAA-SMO, ECC-SMO, DAG-SMO, and BDT-SMO [7, 10].

4. Binary Decision Tree-Based Multiclass SMO (BDT-SMO) Classifiers

It was shown in [11–17] that both OAO-SVM and OAA-SVM have high classification speeds. When the data size is huge, however, the training efficiency will be downgraded severely

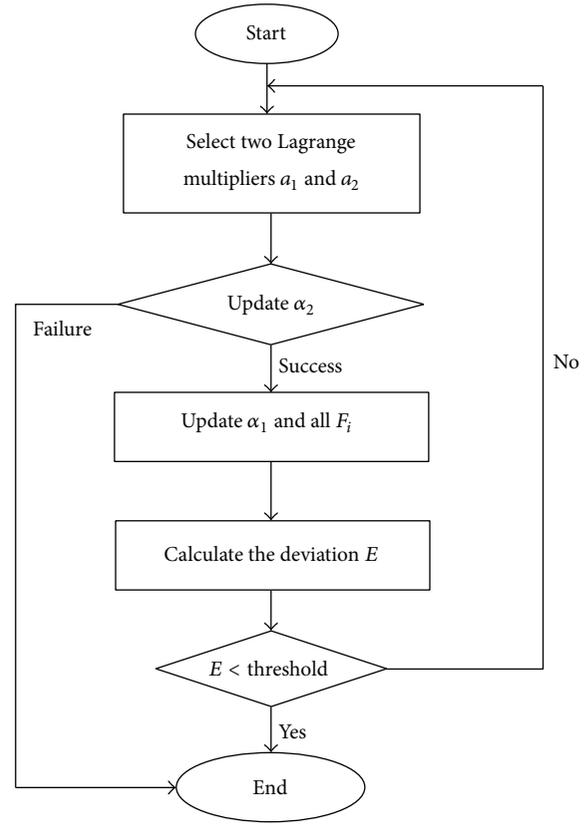


FIGURE 2: Procedures of the SMO training algorithm.

and some regions are inseparable. Thus, they are unsuitable for classifying high-dimensional hyperspectral images that involve large amounts of sample data. ECC-SVM's generalization is independent of the dimensions of the features, but its coding schemes are highly subjective, because the problems of how to determine the code length, generate codes based on the minimum intersymbol Hamming distance, and search for the optimal arrangement have yet to be solved [11, 12]. The DAG-SVM model has a tree structure. Although it can classify quickly and provides an approach to the inseparable region problems, its classification efficiency is unsatisfactory when faced with many classes. In addition, the root nodes are mostly selected empirically, so improperly selected root nodes can directly reduce classification accuracy. BDT-SVM is similar to DAG-SVM, but the difference is that BDT-SVM only needs to construct $n - 1$ two-class SVM classifiers, whereas DAG-SVM needs $n(n - 1)/2$ internal nodes [13]. Additionally, each node in BDT-SVM has only one parent node and the internal node can be something other than a one-against-one classifier. In DAG-SVM, each node can have many parent nodes, and each internal node must be a one-against-one classifier [14]. BDT-SVM is characterized by high classification efficiency, fault tolerance, and generalization ability and thus it can produce better classification results than the other algorithms mentioned above [15].

An improvement on traditional SVM algorithms, SMO classification is faster than traditional algorithms for large

sample datasets. SMO is therefore used in this work to replace traditional SVM algorithms, and the separation-based binary decision tree SMO (BDT-SMO) is employed for classifying hyperspectral remote sensing images. The existing researches [18, 19] have shown that, compared with OAO-SMO, OAA-SMO, ECC-SMO, and DAG-SMO, BDT-SMO has the best accuracy, efficiency, and generalization ability in the classification.

5. Multiple Radial Basis Kernel Function via Weighted Combination

5.1. Definition and Properties of Kernel Function. For the SVM nonlinear classification problem, the input data must be nonlinearly mapped in the original space, R_n , to the linear classifiable data in the high-dimensional feature space using the kernel function, as well as constructing the classification hyperplane in the high-dimensional feature space. If a core function, $K(x_i, x_j)$, satisfies the Mercer conditions, then $K(x_i, x_j)$ corresponds to the inner product in a transformation space [20]. The Mercer theorem was proposed by Hilbert-Schmidt and it states the following: let \mathbf{X} be a compact subset of \mathbf{R}^n , for any given symmetric function, $K(x_i, x_j)$, in $\mathbf{X} \times \mathbf{X} \rightarrow \mathbf{R}$, if its integral operator in the Hilbert space satisfies the integral positive definiteness conditions [20]:

$$\forall f(x) \neq 0, \quad \int f^2(x) dx < \infty, \quad (6)$$

$$\iint K(x_i, x_j) f(x_i) f(x_j) dx_i dx_j > 0;$$

then, there must exist a feature space \mathbf{H} and a mapping $\varphi(x) : \mathbf{X} \rightarrow \mathbf{H}$, and

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j). \quad (7)$$

The kernel function provides an effective approach to SVM nonlinear classification, and the inherent linearity of the high-dimensional space makes SVM more practically feasible. In various practical kernel applications, the kernel function operation in the original input space is equivalent to the high-dimensional operation executed after the nonlinear function is transformed, and the details of the nonlinear mapping are usually unknown. Different kernel functions satisfying the Mercer theorem can be combined as required by the application to construct a combined kernel function of higher complexity, which can be linear or nonlinear [21]. From the kernel function properties, it is known that if $K_1(x_i, x_j)$ and $K_2(x_i, x_j)$ are kernel functions in $\mathbf{R}^n \times \mathbf{R}^n$ and there exist constants $a \geq 0$ and $b \geq 0$, then the function $K(x_i, x_j)$ in the following equations is also the kernel function [21]:

$$K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j),$$

$$K(x_i, x_j) = aK(x_i, x_j), \quad (8)$$

$$K(x_i, x_j) = K_1(x_i, x_j)K_2(x_i, x_j).$$

From the above equations it can be seen that the following equation will hold true if $0 \leq \beta_k < 1$, $\sum_{k=1}^n \beta_k = 1$ and each kernel function $K_k(x_i, x_j)$ satisfies the Mercer conditions above [21]:

$$K(x_i, x_j) = \sum_{k=1}^n \beta_k K_k(x_i, x_j). \quad (9)$$

5.2. Radial Basis Kernel Function. In theory, any function satisfying the Mercer theorem can be used as a kernel function, but different kernel functions must be designed for different applications. Common kernel functions include linear kernels, polynomial kernels, sigmoid kernels, and radial basis functions (RBFs). The first three functions are global kernels, and only RBF is a local kernel. Extensive work has shown that RBF-based SVM outperforms SVM based on the other three kernels and thus is used widely [22]. The inner product of the vectors x and x_i in the feature space for RBF is [23]

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right), \quad (10)$$

where x_i is a center point in RBF, $\|x - x_i\|$ is the norm of the vector $x - x_i$ representing the distance between x and x_i [22], and σ is a width parameter of the function that controls the radial scope of the function and, that is, determines the width of the x_i -centered range within which RBF exerts its influence. RBF is a highly local kernel, and the learning ability of RBF-based SVM is largely dependent on the parameter σ and the punitive factor, C . A small σ value means high SVM learning ability, low empirical risk in the structural risk, and poor generalization ability, giving rise to a large confidence range [24]. SVM uses the punitive factor, C , to balance empirical risk with confidence level to minimize the actual risk. A high value of C means a high degree of data fitting and poor generalization ability. Therefore, proper values of C and σ need to be selected to minimize structural risk and maximize generalization ability for classifiers [25].

5.3. Combined Radial Basis Kernel Function (CRBF). There is some evidence [19, 23] that RBF can achieve high accuracy for spectral and textural feature classification and is capable of multiscale learning. To improve the accuracy and efficiency of spectral remote sensing classification, two different RBFs are firstly used for spectral and textural features and then perform weighted combination to form the CRBF. Consider $[x_i]$ as the feature vector of any pixel in the fused image, $[x_i^s]$ as the combined vector of the pixel's spectral features, $[x_i^w]$ as the pixel's textural feature vector; x_i^s and x_i^w can be combined in a separate or cross manner. The kernels

of the two combination approaches are shown as follows, respectively:

$$K(x_i, x_j) = \sum_{k=1}^m \lambda_k K_k(x_i^s, x_j^s) + \sum_{k=1}^n \lambda_k K_k(x_i^w, x_j^w), \quad (11)$$

$$K(x_i, x_j) = K_s(x_i^s, x_j^s) + K_w(x_i^w, x_j^w) + K_{sw}(x_i^s, x_j^w) + K_{ws}(x_i^w, x_j^s). \quad (12)$$

Equation (11) shows the separate combination of multiple single-kernel RBFs, where λ_k is the weight and m and n are the number of spectral feature kernels and textural feature kernels, respectively. From the kernel properties, it can be seen that if $0 \leq \lambda_k < 1$ and $\sum_{k=1}^n \lambda_k = 1$, then each kernel $K_i(x, x_i)$ satisfies the Mercer conditions. Equation (12) shows the cross-combination of multiple single-kernel RBFs. If each kernel satisfies the Mercer conditions, it needs x_i^s and x_j^w in $K_{sw}(x_i^s, x_j^w)$ and x_i^w and x_j^s in $K_{ws}(x_i^w, x_j^s)$ to have the same number of dimensions. It is seldom used in practice due to the high computational complexity of the kernel. In this work, the spectral features have more dimensions than the textural features, so (11) is necessary for combinatorial classification of spectral and textural features. According to the classification features selected in this work, (11) is adjusted by using the same kernel $K_s(x_i^s, x_j^s)$ for spectral-terrain features. In addition, the same kernel $K_w(x_i^w, x_j^w)$ is used for different types of textural features (gray, gradient, and scale) to reduce the computational complexity of the kernel. The adjusted kernel $K(x_i, x_j)$ is shown as follows:

$$K(x_i, x_j) = \lambda \cdot K_s(x_i^s, x_j^s) + (1 - \lambda) \cdot K_w(x_i^w, x_j^w). \quad (13)$$

The CRBF constructed based on spectral and textural features is shown as follows:

$$K(x_i, x_j) = \lambda \cdot \exp\left(-\frac{\|x_i^s - x_j^s\|}{2\sigma_s^2}\right) + (1 - \lambda) \cdot \exp\left(-\frac{\|x_i^w - x_j^w\|}{2\sigma_w^2}\right), \quad (14)$$

where λ is the weight ($0 \leq \lambda \leq 1$) and σ_s and σ_w are the standardization parameters determining the width of the range surrounding the center of the spectral radial kernel and the center of the textural radial kernel, respectively. The weight λ is used to balance the effect of the spectral radial kernel (SPRBF) and the textural weighted radial basis function kernel (WRBF) in CRBF. When λ is close to 1, SPRBF is dominant in CRBF; otherwise, WRBF is dominant. By adjusting λ , the weights of the effect of spectral and textural features in CRBF can be properly modified, and the optimum combinations to improve the learning and generalization abilities of CRBF can be identified.

6. DT-SMO Parameter Optimization Strategy Based on Genetic Algorithms

The types of kernels and selection of the corresponding parameters have enormous impact on the performance of the SVM classifiers. As mentioned above, the performance of the CRBF-based BDT-SMO classifier is dependent on the punitive factor, C , the weight, λ , and the standardization parameters, δ_s and δ_w . Thus the parameter optimization strategy for CRBF is of great importance. The SVM parameters are traditionally determined by empirical means or via cross-validation. The empirical method is highly subjective and practically infeasible. The cross-validation approach is prone to getting stuck in a local minimum and it is hard to achieve the optimal approximation of a function because it requires the function to be continuous and derivable. In recent years, particle swarm optimization (PSO) algorithms and genetic algorithms (GA) have been used widely and effectively for SVM parameter selection and optimization.

In this work, GA was first used to obtain the kernel parameters of SRBF (including parameters C and δ) and CRBF (including parameters C , λ , δ_s , and δ_w). Next, the trained BDT-SMO classifier is employed to classify the hyperspectral fused images of land usages. SRBF and CRBF parameters can be obtained via GA according to the following steps in Figure 3 [26]. CRBF has more kernel parameters than SRBF, and the parameters optimization processes of SRBF and CRBF are similar, so only the parameters' optimization process of CRBF based on GA is described below.

(1) The BDT-SMO training samples are taken as the original population, the population size is determined, binary encoding is performed on all parameters, and the original GA population is selected randomly.

In order to transfer the excellent gene segments to the next generation better, the corresponding chromosomes (a string composed of specific symbols according to a certain order) of the parameters (C , λ , δ_s , and δ_w) should be designed by a coding mechanism before searching process of GA. In this work, the four parameters of CRBF are encoded with binary codes, and the order of parameters in chromosome is C , λ , δ_s , and δ_w . The lengths of the four parameters are represented with m_1 , m_2 , m_3 , and m_4 , respectively, and the length of the chromosome is $m_1 + m_2 + m_3 + m_4$, so the length of searching interval is $2^{m_1+m_2+m_3+m_4}$. In this work, the values of $m_1 \sim m_4$ are set as follows: $m_1 = 10$, $m_2 = 5$, $m_3 = 10$, and $m_4 = 10$. The searching intervals of the parameters' optimization are set according to the possible value ranges of the parameters as follows: $C \in [0, 1000]$, $\lambda \in [0.01, 1]$, $\delta_s \in [0.01, 10]$, and $\delta_w \in [0.01, 10]$. The structure of the chromosomes is shown in Figure 4.

(2) The fitness function is designed to measure how suitable the selected parameters are, and the fitness of each individual in the population is computed. In this work, the kernel parameters are optimized to improve the prediction accuracy of classification, that is, to make the actual values of the training samples approach the predicted values as closely as possible. Therefore, the total difference between the actual values and the predicted values is a proper measurement of

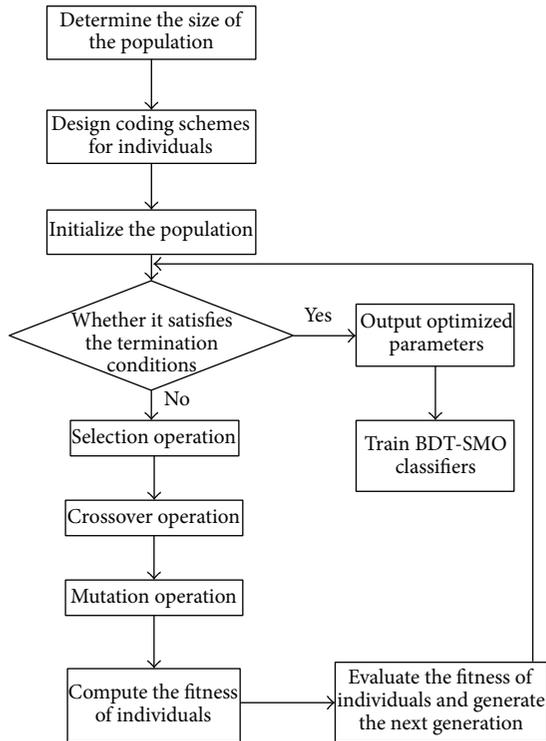


FIGURE 3: GA-based parameter optimization steps for BDT-SMO.

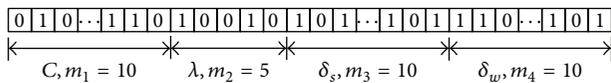


FIGURE 4: The structure of chromosomes.

the fitness. The fitness function used in this work is shown as follows [19]:

$$F(C, \lambda, \delta_s, \delta_w) = 1 - \sqrt{\frac{\sum_{i=1}^n (y_i - y'_i)^2}{n}} \quad (15)$$

In (15), n represents the number of training samples and y'_i and y_i represent the prediction value and actual value of the training sample i . The bigger the F value is, the better the fitness of the corresponding parameters becomes. That is, the quality of the individuals will be better, and the possibility that better genes are inherited to the next generation will be greater.

(3) Independent genetic manipulation is performed on the population: the independent genetic manipulation in this paper includes selection, crossover, mutation, and fitness evaluation. Firstly, the fitness of all individuals is calculated and the roulette wheel selection strategy is adopted to select individuals from the current population for evolution into the next generation. Secondly, a crossover operation is performed using the one-point crossover algorithm and the crossover

probability (P_c) to obtain new individuals for the next generation [27]. Thirdly, a mutation operation is performed using the simple mutation algorithm and the mutation probability (P_m); new individuals are generated by randomly changing some gene segments in the old individuals. Finally, in the fitness evaluation, the fitness function (15) is used to measure the merits and drawbacks of the individuals. In the genetic algorithm, designing the appropriate fitness function can improve the optimization efficiency. In this work, the population size of GA is 100 and the values of P_c and P_m are as follows: $P_c = 0.8$, $P_m = 0.20$.

(4) Check whether it satisfies the termination conditions. If it does, the GA is terminated and the optimal parameters are returned. Otherwise, the third step is repeated to generate new parameter values. When the times of iterations reach the specified number or the satisfied solutions have been obtained, the iterations will be terminated and the results will be outputted. In this work, in order to avoid too long search time, not more than 1000 times of iterations are taken to get the final optimized parameters unless the satisfied values of the four parameters have been obtained.

7. Experiments and Analysis

7.1. Experimental Zones and Samples Selection

7.1.1. *Experimental Zones and Related Images.* To support the credibility of the experimental results, two zones are used in the experiment. The first zone is located in Xindian Village, Jinan District, part of Gulou District, Fuzhou, adjacent to Fuzhou National Forest Park. The covered area forms a rectangular shape, and the positions of the four apexes are (N26°10'24.87", E119°17'40.90"), (N26°10'22.31", E119°20'19.64"), (N26°6'39.45", E119°17'36.49"), and (N26°6'36.90", E119°20'15.14"). The zone has an elevation of 5~500 m, where the northern terrain is higher than the south, and the north is mostly mountainous with an elevation of 50~590 m, covered by laurel forest, mason pine, and tea garden, and interlaced with lanes, shrub-meadow, and bare land. The south part is flat as the core of the urban area of Fuzhou. The EO-1 Hyperion hyperspectral image and the EO-1 ALI panchromatic image for this zone are of the size 155 by 230 pixels and 465 by 690 pixels, respectively, and were captured on March 26th, 2003, covering an area of about 32.09 km².

The second zone is located in Huangyan District, Taizhou, Zhejiang, and is part of the major urban area in Taizhou. The covered area takes a rectangular shape, and the positions of the four apexes are (N28°40'40.14", E121°14'55.85"), (N28°40'42.20", E121°17'37.17"), (N28°37'22.34", E121°14'59.14") and (N28°37'24.40", E121°17'40.37"). The southeastern part of the zone is mostly mountain and forest land with an elevation of 50~500 m, covered by laurel forest, mason pine, and orchard, and interlaced with shrub-meadow, bare land, and lanes. As the heart of the urban area of Fuzhou, other parts of the zone are flat with an elevation of 5~50 m, rich in buildings, farmland, roads, and a small quantity of woodland and rivers. The Hyperion hyperspectral image

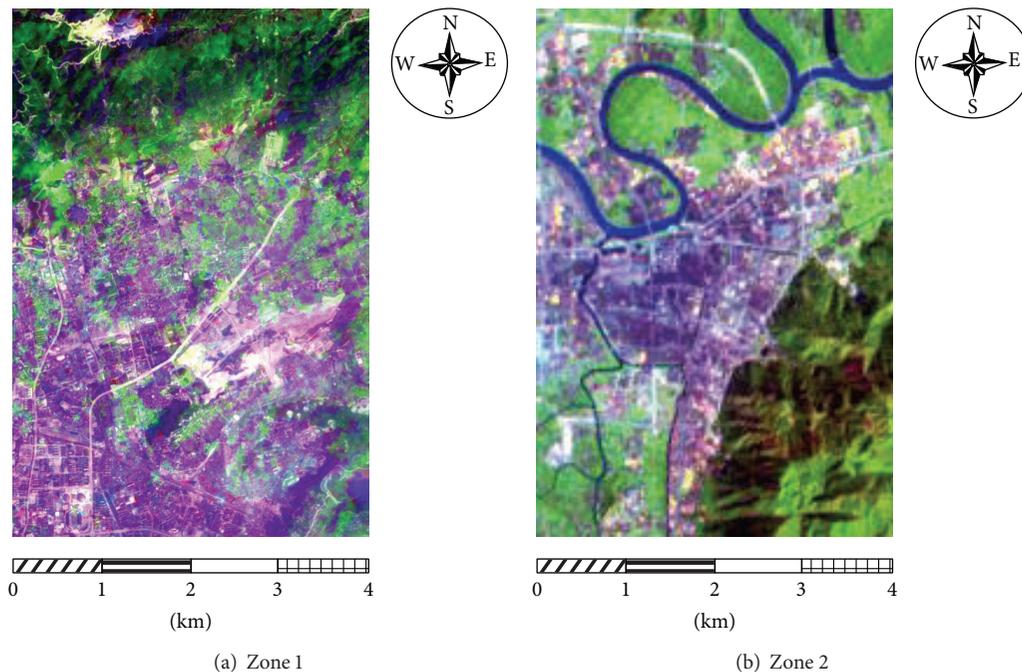


FIGURE 5: Results of fusing images in the two zones.

and the ALI panchromatic image for this zone are 153 by 214 pixels and 459 by 642 pixels, respectively, and were captured on March 10th, 2003, covering an area of about 29.47 km².

The obtained Hyperion hyperspectral images and the ALI panchromatic images needed preprocessing (e.g., remove undesired bands (the original images contain 241 bands), convert radiation values, remove bad lines, repair stripes, eliminate the Smile effect, perform atmospheric correction, geometric accuracy correction, etc.). The aim is to generate Hyperion images with 134 ground reflectance bands and ALI images with 1 panchromatic band. Next, the Gram-Schmidt (GS3) method is adopted to fuse the images above. The images are fused very effectively because the Hyperion and ALI images are both from the EO-1 sensors and have the same phase. In addition, existing research [28] has shown that, compared with other methods of hyperspectral images fusion, Gram-Schmidt (GS3) method can both maintain the spectral features and spatial texture features of hyperspectral images, and it is a relatively ideal method for the fusion of Hyperion hyperspectral images and ALI panchromatic images. The fused images from the two zones are shown in Figure 5.

The land in the two zones is divided into 12 classes based on its usage: farmland (C1), forest (C2), garden (C3), grassland (C4), bare land (C5), industrial and mining warehousing land (C6), residential land (C7), public administration/commerce and service/land concerning foreign affairs (C8), special land (C9), communications and transportation land (C10), water space (C11), and others (C12).

7.1.2. Selection of Samples. Representative regions are selected through field work and comprehensive analysis of remote

sensing images to collect ground-object samples. It is usually required that the training samples are no fewer than the number of test samples and the training samples for each type of ground object are no fewer than the number of classification features. With the assistance of the high resolution images of the zones and the 1:10000 land usage maps, all the representative samples are extracted for each class of land usage from the two experimental zones, as shown in Tables 1 and 2.

After the samples are extracted, it is required that the feature data of the training and test samples are normalized, in order to eliminate the difference of features in terms of dimension, order of magnitude, and data dispersion.

7.2. Extraction and Selection of Classification Features.

Because some ground objects (e.g., roads, buildings, mountains, and rivers) in the experimental zones show highly obvious textural or terrain features, the classification features in this work include the spectral, textural, and terrain features. To improve classification accuracy, as many features as possible are encompassed in this work because the overfitting problem in SVM schemes is not serious when the number of the features is larger.

One hundred and eighty-one features are extracted from the images of the experimental zones, including 136 spectral features, 42 textural features, and 3 terrain features. The spectral features include 134 ground reflectance features, 1 NDVI feature (normalized vegetation index), and 1 NDBI feature (normalized construction index). The textural features involve 6 gray distribution features (second moment, contrast, correlation, variance, inverse gaps, and information entropy extracted through the use of the joint occurrence

TABLE 1: All the samples extracted from the first zone.

Type of land usage	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉	C ₁₀	C ₁₁	C ₁₂
Number of training samples	608	528	838	538	706	616	648	630	604	1294	772	612
Number of test samples	1830	1558	2523	2621	2123	1853	1949	1895	1818	3887	2325	1841

TABLE 2: All the samples extracted from the second zone.

Type of land usage	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉	C ₁₀	C ₁₁	C ₁₂
Number of training samples	810	732	846	720	700	708	848	690	804	1220	806	688
Number of test samples	2435	2203	2540	2163	2105	2127	2545	2073	2415	3664	2423	2071

gray matrix method), 15 gray gradient features [29] (e.g., small gradient dominance, large gradient dominance, heterogeneity of gray distribution extracted via the gray-gradient joint occurrence matrix method proposed by Hong Jiguang in 1984), and 21 textural scale features (semivariance, energy, and mean extracted from the 7 high-frequency components after the second level wavelet decomposition). The terrain features include 3 features (elevation, slope, and aspect) extracted from the DEMs of the experimental zones.

All the features belonging to different types can be represented by different feature vectors as follows:

spectrum: $ST = [ST_1, ST_2, \dots, ST_{134}, ST_{NDVI}, ST_{NDBI}]$;

texture (gray-distribution): $F = [F_1, F_2, \dots, F_6]$;

texture (gray-gradient): $T = [T_1, T_2, \dots, T_{15}]$;

texture (gray-scale): $W = [W_1, W_2, \dots, W_{21}]$;

terrain: $D = [D_{elev}, D_{slp}, D_{asp}]$.

The 5 feature vectors above can be integrated into 16 different combined feature vectors; the details of the different combined feature vectors are shown in Table 3.

In the classification experiment of remote sensing images, the test samples are used to verify the accuracy of the classification, and the classification results can be analyzed through the confusion matrixes. The evaluation indexes of classification results are commonly user accuracy (UA), producers' accuracy (PA), and overall accuracy (OAA) and Kappa coefficient. In addition, the speed of the classification can be measured by the prediction time (TS). In this work, in order to select the best classification method and feature vector, some cross-experiments are performed, respectively, with the 16 different feature vectors above and 5 different classification methods. The classification methods include the minimum distance classifier (MDC), maximum likelihood classifier (MLC), BP neural network (BPN), spectral angle mapping (SAM), and sequential minimal optimization (SMO).

In order to analyze the influence of different numbers of training samples on the classification results, seven different numbers of sample subsets are selected randomly from the samples extracted from the experimental zones. The seven subsets relatively contain 5%, 10%, 20%, 40%, 60%, 80%, and 100% of the extracted samples and are applied to classification experiments using SVM and other methods. The stability of

TABLE 3: The details of the different combined feature vectors.

Number	Combined feature vectors
v1	ST
v2	ST + F
v3	ST + T
v4	ST + W
v5	ST + D
v6	ST + F + T
v7	ST + F + W
v8	ST + F + D
v9	ST + T + W
v10	ST + T + D
v11	ST + W + D
v12	ST + F + T + W
v13	ST + F + T + D
v14	ST + F + W + D
v15	ST + T + W + D
v16	ST + F + T + W + D

the classification results can be measured by the standard deviation. Obviously, each classification set has been repeated 7 times. The experiments are relatively repeated 7 times with different number of samples, and the classification results in Figures 6 and 7 are the stable values in the seven experiments. The experimental results are shown in Figures 6 and 7 (because the classification results of the two experimental zones are very similar, so only the result of experimental zone 1 is shown below).

Figures 6 and 7 show that the accuracy of the classification methods based on only spectral feature vector (v1) is lower than the other feature vectors, and the accuracy and speed of SMO method are higher than the other methods for the all feature vectors. In addition, the accuracy of SMO is the highest of all using the feature vector of v12 (i.e., ST + F + T + W).

In addition, the different combined feature vectors have the similar effects on the Kappa coefficient and the overall accuracy, so the effect on the latter will not be shown here.

Therefore, in the land use classification of hyperspectral fused images, the method of SMO based on the features (178 dimensions) combined of spectrum and textural features can achieve the best classification accuracy and speed.

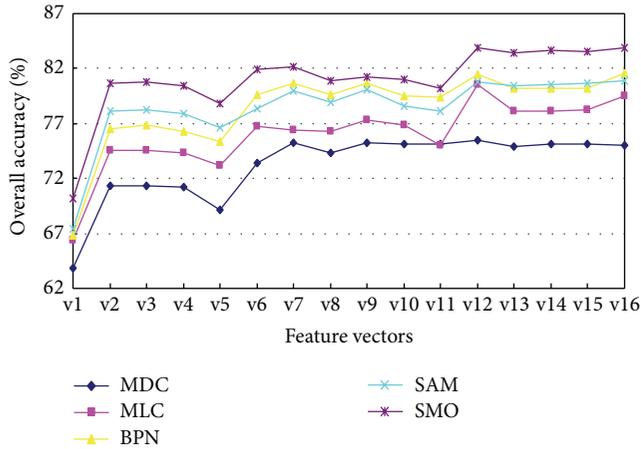


FIGURE 6: The effect on overall classification accuracy by different feature vectors and classification methods (experimental zone 1).

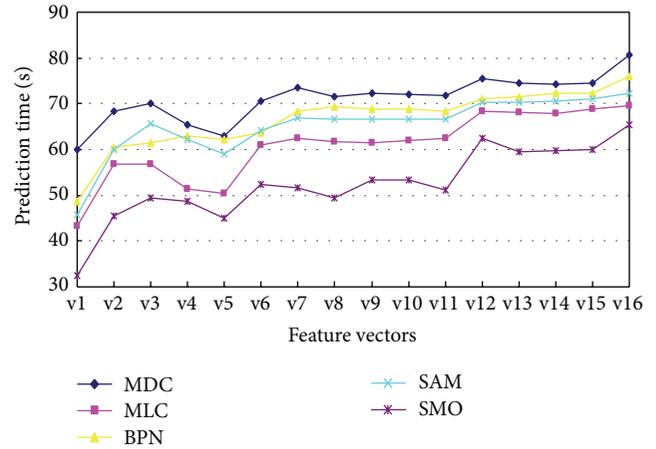


FIGURE 7: The effect on classification prediction time by different feature vectors and classification methods (experimental zone 1).

7.3. *SMO-Based Multiclass Classification of Hyperspectral Fused Images of Land Usage.* It is first required that the optimal SMO-based multiclass classification approach is determined before classification using the combined kernel-based SMO method which is undertaken. The training samples of the two zones are learned using OAO-SMO, OAA-SMO, DAG-SMO, ECC-SMO, and BDT-SMO, respectively, based on the same kernel (SRBF). GA is used to optimize the kernel parameters C and δ . The experiments are relatively repeated 7 times with different number of samples like in Section 7.2. The seven subsets relatively contain 5%, 10%, 20%, 40%, 60%, 80%, and 100% of the extracted samples and are applied to classification experiments using different SMO multiclass classification approaches. The stability of the classification results can be measured by the standard deviation. The experimental results are provided in Tables 4 and 5, and the classification accuracies in tables are all the stable values in the seven experiments.

From Tables 4 and 5, the overall accuracy, average accuracy, Kappa coefficient values, and standard deviation of BDT-SMO approach are all higher than the other approaches. Obviously, BDT-SMO is the best multiclass classification approach for hyperspectral fused images.

7.4. *CRBF-Based BDT-SMO Classification of Hyperspectral Fused Images of Land Usage.* Land usages of the two zones are classified using linear, polynomial, sigmoid, SRBF, and CRBF-based BDT-SMO classifiers, respectively. The experiments are relatively repeated 7 times with different number of samples like in Section 7.3. The seven subsets relatively contain 5%, 10%, 20%, 40%, 60%, 80%, and 100% of the extracted samples and are applied to BDT-SMO classification experiments using different kernel functions. The stability of the classification results can be also measured by the standard deviation. The experimental results are provided in Tables 6 and 7, and the classification accuracies in tables are all the stable values in the seven experiments.

Tables 6 and 7 deliver the experimental results of the two zones, where r and d denote the offset coefficient and the

number of polynomial orders, v and w denote the scale and attenuation parameters of the sigmoid kernel, and GA are employed to optimize the kernel parameters.

From Tables 6 and 7, it can be seen that the CRBF-based BDT-SMO classifiers have the highest overall accuracy (at 85.43% and 86.15%, 1.97% and 2.10% higher than the SRBF kernel), average accuracy (at 85.67% and 85.52%, 2.01% and 2.15% higher than the SRBF kernel), and Kappa coefficient values (0.8411 and 0.8483). The standard deviations of the classification accuracies of the CRBF-based BDT-SMO classifiers (0.63 and 0.62) are obviously lower than the others. In addition, CRBF has the fastest training speed second only to SRBF and the fastest test speed. The standard deviations of the CRBF-based and SRBF-based BDT-SMO classifiers are very similar and obviously lower than the others. Although, compared with SRBF, the training speed of the CRBF-based BDT-SMO classifiers is a little lower than SRBF-based, but the CRBF-based BDT-SMO classifiers have the better accuracy and test speed in classification and have almost the same stability with the SRBF-based BDT-SMO classifiers. Apparently, CRBF-based BDT-SMO classifiers have the better performance. The results of the CRBF-based and SRBF-based BDT-SMO classification of hyperspectral sensing images of land usages are provided in Figures 8 and 9. In addition to the overall classification results, the accuracy of each individual class of land usages is also improved, as shown in Tables 8 and 9. From the tables above, it can be seen that CRBF can more accurately classify each class of land usage. Especially for the four difficult classes— C_3 (garden), C_5 (bare land), C_8 (public administration/commerce and service/land concerning foreign affairs), and C_{10} (communications and transportation land), the classification accuracy is improved enormously, and the producers' accuracy (PA) is up to 80.04%–80.51%, 66.17%–67.84%, 83.13%–84.76%, and 82.54%–84.07%, which is 3.99%–5.93%, 2.26%–2.78%, 3.37%–5.21%, and 3.13%–4.44% higher than the single RBF kernel. Therefore, the CRBF-based BDT-SMO classifiers can classify the land usages using hyperspectral fused images in experimental zones more effectively.

TABLE 4: Comparison of classification results under different SMO multiclass classification approaches (zone 1).

Classification approaches	Overall accuracy (%)	Average accuracy (%)	Kappa coefficient	Training time (s)	Test time (s)	Standard deviation	Optimized parameters
OAO-SMO	82.43	80.06	0.7998	24.36	21.35	0.95	$C = 10, \delta = 0.075$
OAA-SMO	81.09	79.89	0.7875	25.37	24.10	1.85	$C = 10, \delta = 0.070$
DAG-SMO	81.25	81.29	0.7904	30.42	15.70	1.36	$C = 15, \delta = 0.050$
ECC-SMO	82.05	82.34	0.7954	31.71	15.05	1.65	$C = 10, \delta = 0.550$
BDT-SMO	83.46	83.66	0.8195	30.57	13.67	0.65	$C = 15, \delta = 0.055$

TABLE 5: Comparison of classification results under different SMO multiclass classification approaches (zone 2).

Classification approaches	Overall accuracy (%)	Average accuracy (%)	Kappa coefficient	Training time (s)	Test time (s)	Standard deviation	Optimized parameters
OAO-SMO	81.57	81.79	0.7994	32.33	24.74	1.11	$C = 10, \delta = 0.080$
OAA-SMO	81.21	80.75	0.7837	31.64	26.75	1.21	$C = 10, \delta = 0.075$
DAG-SMO	82.31	80.79	0.8106	40.12	16.84	1.32	$C = 15, \delta = 0.055$
ECC-SMO	82.52	81.32	0.8153	43.35	16.01	1.33	$C = 10, \delta = 0.550$
BDT-SMO	84.05	83.37	0.8254	42.11	14.59	0.59	$C = 15, \delta = 0.065$

TABLE 6: Comparison of BDT-SMO classification approaches under different kernels (zone 1).

Kernels	Overall accuracy (%)	Average accuracy (%)	Kappa coefficient	Training time (s)	Test time (s)	Standard deviation	Optimized parameters
Linear	79.34	79.39	0.7752	35.24	18.49	1.88	
Polynomial	80.05	80.41	0.7841	39.79	20.63	1.79	$r = 1, d = 3$
Sigmoid	82.12	82.55	0.8024	38.15	22.94	1.52	$\nu = 20, w = 0.50$
SRBF	83.46	83.66	0.8195	30.57	13.67	0.65	$C = 15, \delta = 0.055$
CRBF	85.43	85.67	0.8411	33.27	12.44	0.63	$C = 15, \lambda = 0.255$ $\delta_s = 0.035, \delta_w = 0.055$

TABLE 7: Comparison of BDT-SMO classification approaches under different kernels (zone 2).

Kernels	Overall accuracy (%)	Average accuracy (%)	Kappa coefficient	Training time (s)	Test time (s)	Standard deviation	Optimized parameters
Linear	80.11	79.33	0.7841	43.15	16.41	1.85	
Polynomial	81.19	80.06	0.7859	43.59	22.79	1.75	$r = 1, d = 3$
Sigmoid	82.49	81.39	0.8009	44.95	15.29	1.61	$\nu = 20, w = 0.55$
SRBF	84.05	83.37	0.8254	42.11	14.59	0.59	$C = 15, \delta = 0.065$
CRBF	86.15	85.52	0.8483	43.14	13.95	0.62	$C = 15, \lambda = 0.255$ $\delta_s = 0.045, \delta_w = 0.055$

TABLE 8: Land usage classification accuracy (zone 1).

Type of land usage	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉	C ₁₀	C ₁₁	C ₁₂
Zone 1 (%)	92.13	95.67	76.05	88.45	63.91	82.86	84.89	79.76	88.51	79.41	91.90	80.36
Zone 2 (%)	90.91	97.58	74.58	84.77	65.06	83.40	84.18	79.55	88.28	79.63	91.03	81.46

TABLE 9: Land usage classification accuracy (zone 2).

Type of land usage	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉	C ₁₀	C ₁₁	C ₁₂
Zone 1 (%)	93.70	99.78	80.04	89.44	66.17	82.16	86.15	83.13	89.08	82.54	93.48	82.34
Zone 2 (%)	92.38	99.57	80.51	85.55	67.84	82.21	84.32	84.76	89.24	84.07	92.03	83.71

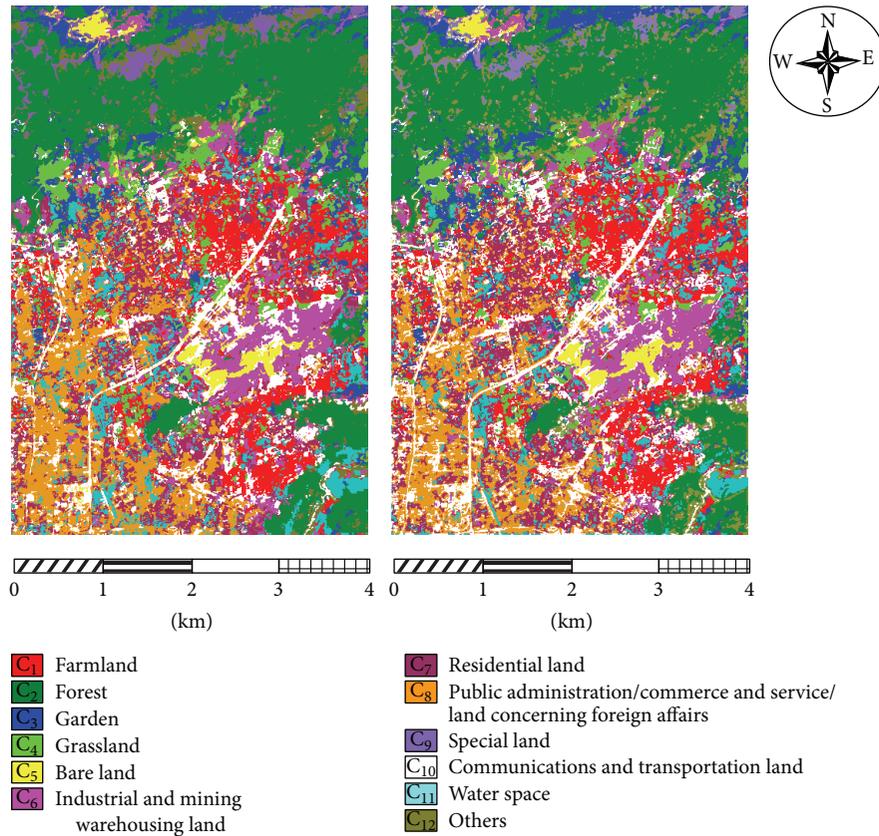


FIGURE 8: Comparison of BDT-SMO classification based on SRBF and CRBF (zone 1).

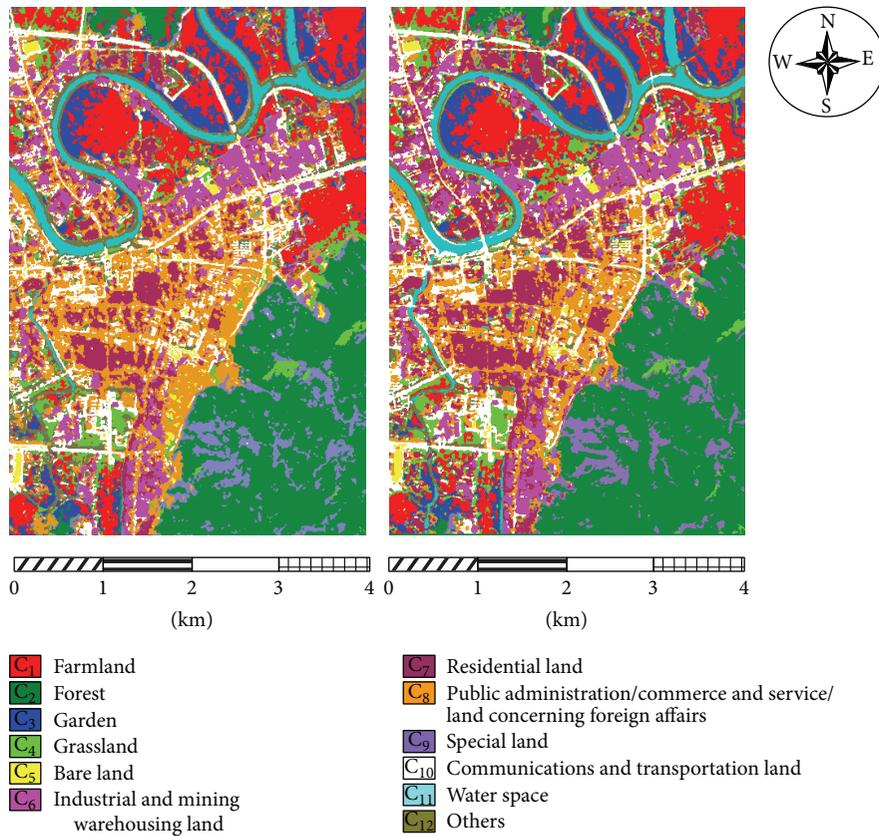


FIGURE 9: Comparison of BDT-SMO classification based on SRBF and CRBF (zone 2).

TABLE 10: CRBF-based BDT-SMO classification examples.

Verification instance	Zone	Specific position	Right class	Classification results (SRBF)	Classification results (CRBF)
Instance 1	Zone 1	Near Jiao Mountain in the north of Jin'an District	C_3	C_2	C_3
Instance 2	Zone 2	Between MaAnShan Village and WangLinYang Village and nearby	C_3	C_1	C_3
Instance 3	Zone 1	Near Fontainebleau Community and the Residence Theme Park	C_5	C_{10}	C_5
Instance 4	Zone 2	Near YaLin Village in the south Second Ring Road	C_5	C_{10}	C_5
Instance 5	Zone 1	Near JianTianJiuDong Innovation Garden in the north of XiFeng Road	C_8	C_{10}	C_8
Instance 6	Zone 2	Near the First Vocational and Technical College in HuangYan	C_8	C_{10}	C_8
Instance 7	Zone 1	Section of the South FuFei Road between the Third Ring freeway and the North Ring Road, and the middle section of the North Ring Road	C_{10}	C_8/C_5	C_{10}
Instance 8	Zone 2	YongNingJiang Bridge (bridge floors of the East Second Ring Road, DaQiao Road and LuTing Road)	C_{10}	C_{12}	C_{10}

To more intuitively show the benefit of CRBF for classification of C_3 , C_5 , C_8 , and C_{10} , 4 instances are selected, respectively, from the two zones for verification. BDT-SMO classification results of these instances under the single RBF and CRBF are provided in Table 10.

From Table 10, it can be seen that the 8 positions are all misclassified when the SRBF-based BDT-SMO classifier is used and are all classified correctly when the CRBF-based BDT-SMO classifier is used. Therefore, the CRBF-based BDT-SMO classifier is excellent enough to classify land usage using hyperspectral fused images effectively and efficiently.

8. Conclusions

In order to improve the accuracy and efficiency of the classification of hyperspectral fused images, a new combined radial basis kernel function (CRBF) is proposed by combining spectral and textural features in a weighted manner. The binary decision tree-based multiclass SMO (BDT-SMO) based on the kernel function of CRBF is used in the classification of hyperspectral fused images, and the parameters of CRBF-based BDT-SMO classifiers are optimized by genetic algorithms (GA). Experimental results show that, compared with SRBF, CRBF-based BDT-SMO classifiers have better accuracy and efficiency in the classification of hyperspectral fused images.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (no. 41201417) and the Science and Technology Project of Fujian Province Education Department of China (no. JA13364). The authors would like to thank Zhilei Lin and Professor Luming Yan for useful assistance, suggestions, and discussions.

References

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [2] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [5] G. M. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification," *Remote Sensing of Environment*, vol. 93, no. 1-2, pp. 107–117, 2004.
- [6] G. M. Foody and A. Mathur, "The use of small training sets containing mixed pixels for accurate hard image classification: training on mixed spectral responses for classification by a SVM," *Remote Sensing of Environment*, vol. 103, no. 2, pp. 179–189, 2006.
- [7] N. Xue, "Comparison of multi-class support vector machines," *Computer Engineering and Design*, vol. 32, no. 5, pp. 1792–1795, 2011.
- [8] T. N. Do, V.-H. Nguyen, and F. Poulet, "GPU-based parallel SVM algorithm," *Journal of Frontiers of Computer Science and Technology*, vol. 3, no. 4, pp. 368–377, 2009.

- [9] J. C. Platt, *Fast Training of support Vector Machines Using Sequential Minimal Optimization*, MIT Press, Cambridge, Mass, USA, 1999.
- [10] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.
- [11] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, no. 2, pp. 263–286, 1995.
- [12] Y. Liu, "Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus," in *Proceedings of the 9th International Conference on Spoken Language Processing, INTERSPEECH (ICSLP '06)*, pp. 1938–1941, Pittsburgh, Pa, USA, September 2006.
- [13] Q. Ai, Y. Qin, and J. Zhao, "An improved directed acyclic graphs support vector machine," *Computer Engineering and Science*, vol. 33, no. 10, pp. 145–148, 2011.
- [14] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Proceedings of Advances in Neural Information Processing Systems*, vol. 12, pp. 547–553, 2000.
- [15] K. Li, Q. Ren, B. Wen et al., "Application of separation degree and binary decision tree-based multi-class SVM in classification and recognition of weld defects," *Journal of Sichuan University (Natural Science)*, no. 3, pp. 520–524, 2010.
- [16] X. Wang and Y. Qin, "Research on SVM multi-class classification based on binary tree," *Journal of Hunan Institute of Engineering*, vol. 18, no. 3, pp. 68–70, 2008.
- [17] G. Madzarov and D. Gjorgjevik, "Multi-class classification using support vector machines in decision tree architecture," in *Proceeding of the IEEE EUROCON 2009 (EUROCON '09)*, pp. 288–295, St.-Petersburg, Russia, May 2009.
- [18] F. Huang, "Research on classification of hyperspectral remote sensing imagery based on BDT-SMO and combined features," *Journal of Multimedia*, vol. 9, no. 3, pp. 456–462, 2014.
- [19] T. R. Tooke, N. C. Coops, N. R. Goodwin, and J. A. Voogt, "Extracting urban vegetation characteristics using spectral mixture analysis and decision tree classifications," *Remote Sensing of Environment*, vol. 113, no. 2, pp. 398–407, 2009.
- [20] G. Wang, "Properties and construction methods of kernel in support vector machine," *Computer Science*, vol. 33, no. 6, pp. 172–178, 2006.
- [21] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Mari, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, 2006.
- [22] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.
- [23] N. Benoudjit and M. Verleysen, "On the kernel widths in radial-basis function networks," *Neural Processing Letters*, vol. 18, no. 2, pp. 139–154, 2003.
- [24] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.
- [25] G. Feng, "Parameter optimizing for Support Vector Machines classification," *Computer Engineering and Applications*, vol. 47, no. 3, pp. 123–124, 2011.
- [26] C. Huang and C. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231–240, 2006.
- [27] M. Zhong, "Research on intelligent schedule of public traffic vehicles based on parallel genetic algorithm," *Computer Era*, no. 12, pp. 18–20, 2011.
- [28] F. Huang and L. Yan, "Study on the hyperspectral image fusion based on the gram-schmidt improved algorithm," *Information Technology Journal*, vol. 12, no. 22, pp. 6694–6701, 2013.
- [29] J. Hong, "The texture analysis methods of gray gradient co-occurrence matrix," *Acta Automatica Sinica*, vol. 10, no. 1, pp. 22–25, 1984.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

