

Research Article

A New Semisupervised-Entropy Framework of Hyperspectral Image Classification Based on Random Forest

Mengmeng Sun,¹ Chunyang Wang ,^{1,2} Shuangting Wang,^{1,2} Zongze Zhao,^{1,2} and Xiao Li¹

¹School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China

²Henan Province Engineering Technology Research Center of Space Big-Data Acquisition Equipment Development and Application, Henan Polytechnic University, Jiaozuo 454000, China

Correspondence should be addressed to Chunyang Wang; hpu_wcy@163.com

Received 26 May 2018; Accepted 14 August 2018; Published 4 September 2018

Academic Editor: Lei Zhang

Copyright © 2018 Mengmeng Sun et al. 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.

The purposes of the algorithm presented in this paper are to select features with the highest average separability by using the random forest method to distinguish categories that are easy to distinguish and to select the most divisible features from the most difficult categories using the weighted entropy algorithm. The framework is composed of five parts: (1) random samples selection with (2) probabilistic output initial random forest classification processing based on the number of votes; (3) semisupervised classification, which is an improvement of the supervision classification of random forest based on the weighted entropy algorithm; (4) precision evaluation; and (5) a comparison with the traditional minimum distance classification and the support vector machine (SVM) classification. In order to verify the universality of the proposed algorithm, two different data sources are tested, which are AVIRIS and Hyperion data. The results show that the overall classification accuracy of AVIRIS data is up to 87.36%, the kappa coefficient is up to 0.8591, and the classification time is 22.72s. Hyperion data is up to 99.17%, the kappa coefficient is up to 0.9904, and the classification time is 8.16s. Classification accuracy is obviously improved and efficiency is greatly improved, compared with the minimum distance and the SVM classifier and the CART classifier.

1. Introduction

As shown in Figure 1, hyperspectral remote sensing image technology contains a lot of potential information and integrates the spectral and spatial dimensions [1, 2]. It has the characteristics of a continuous spectrum and a unified spectrum, and it can be used to interpret an object with high spectral diagnosis ability [3]. Considering these advantages, we use hyperspectral remote sensing image in the experiments in this paper.

Owing to the limitations of technology, the mining of spatial dimension information is obviously deficient. How to fully excavate a large amount of information hidden in hyperspectral remote sensing images is a key issue in the literature. To be of scientific merit, classification technology is the important technology for processing hyperspectral remote sensing images [4]. Using hyperspectral images to classify ground objects is one of the core contents of the application of hyperspectral remote sensing technology [5]

and the classification results have great application value that may apply to land cover [6, 7], resource surveys [8–12], environment monitoring [13, 14], coverage prediction [15, 16], military exploration [17, 18], and other fields.

However, in the process of application, we mainly encounter the problems of the Hughes phenomenon, Bellman's disaster, nonlinear distribution of data in feature space, and so on, which may result in the information being ambiguous during the process of hyperspectral image classification [19]. What is more, it is really important to fine-tune the spectral features provided by spectral images considering that the traditional classification methods based on a single classifier cannot meet the classification needs of hyperspectral remote sensing [20] and most of the traditional algorithms take pixels as the basic unit for classifying [21, 22], without considering the spatial features of remote sensing images. This results in the algorithm not being able to deal with the "isomorphism problem" of the same objects effectively [23], and many noise points easily appear in the

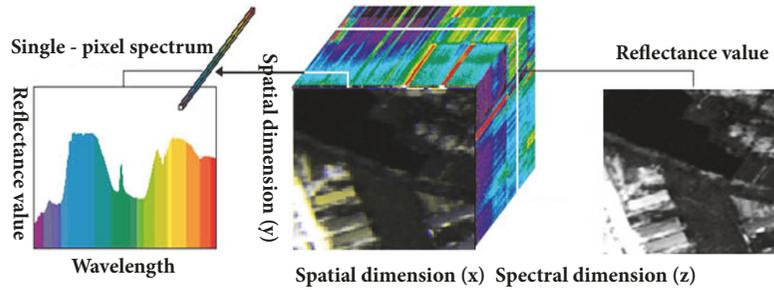


FIGURE 1: The example of hyperspectral image data cube.

interior of the ground objects in the classification results. The fine spectral features provided by spectral images can be used to distinguish objects with subtle differences, including those with high similarity to natural backgrounds, and the distribution of background information is different from the assumption of the model. The object size is affected by the subpixel level, and sometimes the false alarm rate is too high [24]. Therefore, hyperspectral image target detection technology has great potential value in the field of public security and national defense. Hyperspectral image target detection requires the diagnostic spectral characteristics of a target, and it is applied to many varieties of target spectrum in practice [25]. Therefore, it is necessary to develop a stable and reliable method.

To be of scientific merit, it is necessary to make full use of the rich space and spectral information of hyperspectral remote sensing images to interpret the objects of observation. This has become a hotspot in the research field and is the frontier field in recent years. What is more, it has great application value and broad development prospects in many related fields. As has been illustrated, it is essential to improve the extraction ability of ground object information [26].

In view of the characteristics of hyperspectral remote sensing images, the classification of random forest algorithms may be a good choice. The random forest algorithm is a supervised self-training classifier [27–30] that consists of many classification trees, each of which completes its own sorting operation. The final classification results are determined by the voting results of each classification tree [31]. The random forest algorithm is a classification method based on the principle of the classification and regression tree (CART) decision trees algorithm, which is composed of a series of CART decision trees. The classification results are voted on by the decision trees. The final feature category relies on the classification result with the largest number of votes [32]. The Gini index is used to measure the classification results.

As an excellent classifier model, random forest also provides a new idea for image classification [33]. Random forest can connect independent variables with dependent variables by generating a lot of classification trees, which can successfully calculate the nonlinear and interactive effects of variables, even when there is a high degree of interference. Considering those characteristics, random forest is a good choice that can deal with the problems of local extremum in

hyperspectral remote sensing images, the difference between different categories of ground object, and the slow speed of running the operation. Many explanatory variables can be predicted [34]. This algorithm is a summary of classification trees and gives importance to all variables. However, it is still relatively robust in the face of data loss and imbalance.

In view of the large amount of information contained in hyperspectral remote sensing images and the fact that they are still a new subdivision of spectral imaging remote sensing technology, it is very difficult to fully exploit the potential information contained therein and to remove the difficulty in obtaining training samples. Some of the traditional supervised classification methods are not very practical; however, the unsupervised classification method does not require training samples owing to the limitation of its classification accuracy. Owing to the above reasons, it may be wise to use semisupervised classification to classify ground objects [13]. Semisupervised classification is a learning algorithm that is an active learning machine. Integrated learning methods are a very important research direction in the field of machine learning. Semisupervised classification is an active learning algorithm that focuses on the use of labeled and unlabeled samples [35] to obtain high-performance classifiers. The purpose of ensemble learning is to improve the accuracy of weak learning classifiers by integrating multiple learning devices. Semisupervised ensemble learning is a new machine learning method that combines semisupervised learning and integrated learning to improve the generalization performance of classifiers [36].

Semisupervised learning uses both labeled and unlabeled samples in the process of training. With the growth of information, the classification problem becomes more and more complicated, while the semisupervised classification algorithm obtains only a small number of classification samples. A small number of labeled samples are used to train the classification model in this classification algorithm.

Semisupervised learning is a self-training machine learning method that makes full use of a small number of labeled samples and a large number of unlabeled samples. In view of the high cost of sample marking, the rapid development of spectral imaging technology, and the emergence of hyperspectral images, it is very meaningful to study the classification method of semisupervised machine learning. This invention can optimize the classification performance of hyperspectral remote sensing images and improve their classification accuracy and efficiency.

Random forest not only shows high classification performance but also has fewer parameters to be adjusted, and it is fast and efficient in the field of machine learning [37]. There is no need to worry about overfitting and strong noise tolerance [38]. Excellent random performance makes it widely used in intelligent information processing, bioinformatics, finance, diagnosis of faults, recognition of images, industrial automation, and other fields [39]. It has attracted widespread attention and achieved great success. Industrial automation and other fields have widely used it and achieved great success, attracting extensive attention [37, 40]; although many scholars have conducted extensive research on random forests and achieved many remarkable things [41], there are still some limitations and shortcomings, leaving some room for improvement [16]. Therefore, in order to deal with the problems of the high dimensionality and large amount of data of hyperspectral remote sensing images and the difficulty in extracting samples' extraction characteristics [42, 43], a robust classification method with high accuracy is urgently needed [44].

In this paper, a semisupervised random forest hyperspectral remote sensing image classification method based on weighted entropy [45–49] is proposed. It can be classified by randomly selecting the number of training samples with 10% or 5% labeled samples. A classifier model [50–52] is constructed that uses a random forest based on a CART decision tree with probabilistic output; it is a supervised classifier [53] that integrates multiple weak classifiers. The classifier predicts ground objects according to the number of votes cast. Then, a weighted entropy algorithm is used to give the class with the highest weighted values and return weight values, which is predicted by the model. The objects with larger weighted entropy values are sorted, and objects that account for 5% or 10% of the total sample increase are added to the training samples to form new training samples; then, the prediction and classification steps are carried out again. The above steps are repeated until the conditions for iteration stop are satisfied or the labeled samples are used up; the remaining samples are used for accuracy evaluation, which uses classifier performance detection [50, 54].

This classification method is economical and suited to the properties of hyperspectral remote sensing images. It is of high value to researchers wishing to classify large, dense areas. The purpose of the algorithm presented in this paper is to select the features with the highest average separability by using the random forest method to distinguish the categories that are easy to distinguish and to select the most divisible features of the most difficult categories by using the weighted entropy algorithm. The framework is composed of five parts: (1) random samples selection with (2) probabilistic output initial random forest classification processing based on the number of votes; (3) semisupervised classification, which is the improvement of the supervision classification of random forest based on the weighted entropy algorithm; (4) precision evaluation; and (5) a comparison with the traditional minimum distance classification and SVM classification. In order to verify the universality of the proposed algorithm, two different data sources are tested: AVIRIS and Hyperion data. The results show that the overall

classification accuracy of AVIRIS data is up to 87.36%, the kappa coefficient is up to 0.8591, and the classification time is 22.72s. The Hyperion data is up to 99.17% accurate, the kappa coefficient is up to 0.9904, and the classification time is 8.16s.

2. Materials and Methods

The imaging spectrometer acquiring the hyperspectral image data cannot be directly applied and classified analysis, which needs to be analyzed. Therefore, the preprocessing of hyperspectral remote sensing image in general includes atmospheric radiation correction, geometry correction, and noise removal [45, 55–57]. In the preprocessing of hyperspectral image, radiometric correction is the mainly steps.

In this experiment, there are three classification methods adapted, which are the minimum distance classification method, the support vector machine classification method, and the semisupervised classification method proposed in this paper which uses the features of the random samples and random band selection [58] of random forest based on the weighted entropy [31]. The semisupervised classifier [59] is trained with labeled samples data and determined the parameters of the classifier until it ensures that the training samples are matched with the verification samples. In the case of mutual independence, the fitting test of classification parameters is carried out to determine the applicability of the parameters [60]. The minimum distance classifier and support vector machine classifier are used to test the function of these classifiers with the same number of training samples. The accuracy evaluation is carried out to quantitatively evaluate results of the experimental method to determine the effect of the classifiers.

In order to verify the universality of the proposed algorithm, two different data sources are tested in this paper. In fact, there will be three parts in this chapter. First of all, the study areas and the training samples will be illustrated. In the second part, the selection of classification algorithms will be illustrated. In the third part, the processing of the algorithm that is proposed in the paper will be illustrated.

2.1. Study Areas and the Training Samples. In order to verify the universality of the proposed algorithm, two different data sources are tested in this paper, which are AVIRIS data and Hyperion data. The AVIRIS imaging spectrometer, also known as the airborne visible infrared imaging spectrometer, was developed in 1987 by NASA's Jet Propulsion Laboratory (JPL). It covers a wavelength range of 400,500 nm and it is almost the full wavelength of solar radiation. Because of its rich spectral information, AVIRIS provides a large amount of data for various science and applications. The images used in this paper are located at the Kennedy Space Center in Florida, USA. The acquisition time is March 1996. The image is 614 pixels wide and 512 pixels high, with a total of 15 pixels five bands with a spectral resolution of 10 nm and a spatial resolution of 18 meters. The training data are selected on the basis of images provided by Landsat Thematic Mapper.

The land cover types in the region are divided into 13 major categories, namely, scrub, willow, CP-Hammock,

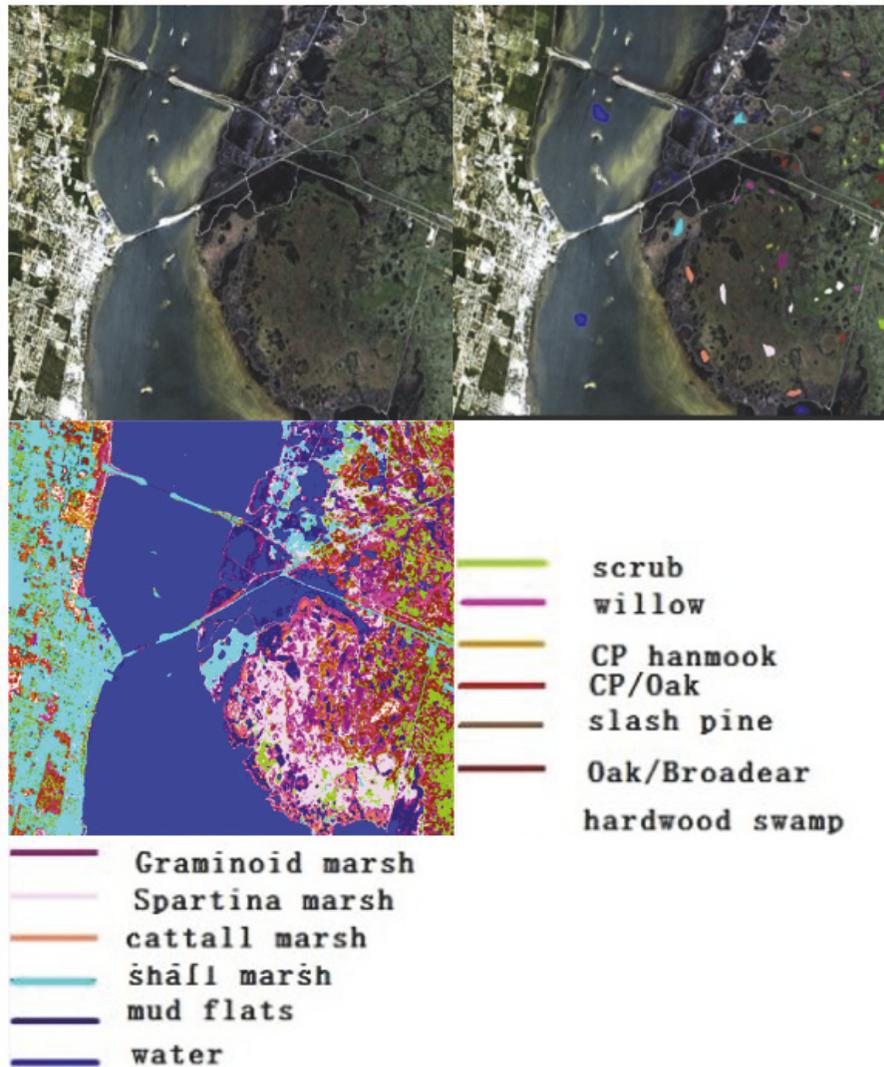


FIGURE 2: The synthesis true color image (bands 31, 21, and 11) and the distribution of training samples of AVIRIS data and the ground truth data.

CPP/Oak, slash-Pine, Oak/Broad-leaf, Hardwood, Graminoid-marsh, Spartina-marsh, Cattail marsh, Saltmarsh Mud flats, and water. In the classification of remote sensing, the selection of the training samples area directly determines the classification results. Using Google Earth to select the training samples area of remote sensing classification is an excellent way. The ground truth data can be used in two ways: one is the standard classification diagram and the other is the selected area of interest (validation samples area). In this paper, the standard classification diagram is used as the ground truth data. The true color composites images of different samples [27]. AVIRIS data, the distribution of training samples, and the ground truth data are shown in Figure 2.

The Hyperion imaging spectrometer, mounted on the EO-1 satellite platform, was launched by NASA on November 2000. It covers a wavelength range of 400 to 2500 nm, has 220 wavelengths, and has a spectral resolution of 10 nm and a spatial resolution of 30 meters. The image used in

this paper is located in Dali city, Yunnan province, China. This product was created by US Geological Survey. The product contains EO-1 Hyperion data file, hierarchical data format, or geographical mark image file format (TIFF). EO-1 has launched a one-year technical demonstration validation mission. NASA (NASA) and the US Geological Survey (USGS) agreed to continue the EO-1 program as an extended mission. Information about EO-1 satellites and Hyperion sensors can be found at the USGS and NASA's Web site: <http://eo1.usgs.gov> <http://eo1.gsfc.nasa.gov>. The date of acquisition of this image is January 2014. In order to use it conveniently, this paper cuts out an experimental area of 529 pixels high and 256 pixels wide. After atmospheric correction and geometric correction, a total of 72 bands were selected for analysis after removing the low SNR band. The images of land cover mainly include bare land, low residents, low vegetation, broad-leaved forest, and six types of water body. In the classification of remote sensing, the selection of the

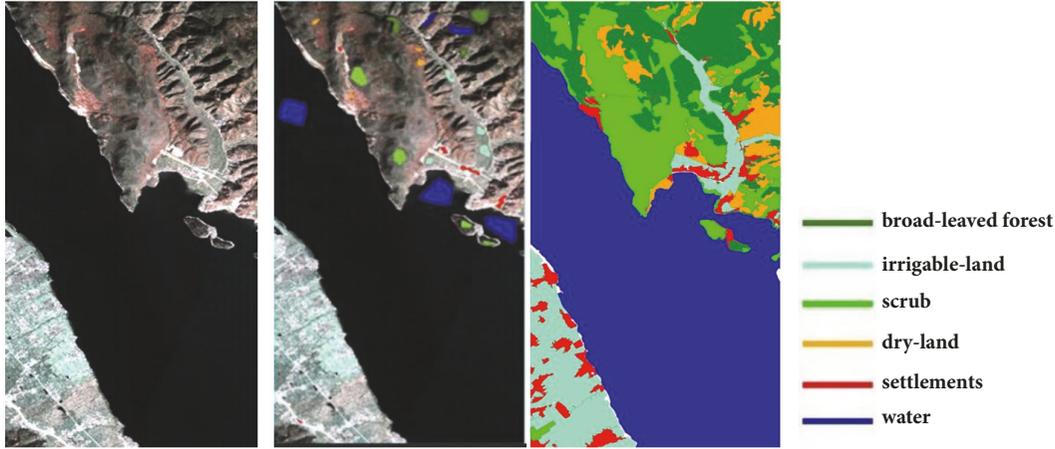


FIGURE 3: The synthesis true color image (bands 29, 20, and 18) and the overlaid image of training sets of Hyperion data and the ground truth data.

training samples area directly determines the classification results. Using Google Earth to select the training samples area of remote sensing classification is an excellent way. The ground truth data can be used in two ways: one is the standard classification diagram and the other is the selected area of interest (validation samples area). In this paper, the standard classification diagram is used as the ground truth data. Hyperion data true color composite image and the distribution of training samples and the ground truth data are shown in Figure 3.

Classification is one of the main problems in the field of remote sensing. As a tool to solve the problem, classifier is always a hot topic. The commonly used classifiers include decision tree, logical regression, Bayes, and neural network. These classifiers all have their own performance characteristics. In this paper, three kinds of classifiers are mainly used, that is, the traditional classifier based on minimum distance and the support vector machine classifier and the classifier proposed in this paper.

The essence of classification is to select the appropriate discriminant function according to the law of probability and statistics and to establish a reasonable discriminant model to separate the discrete clusters in the image and to make the judgment and classification. Through the statistics and calculation of the regions of interest, the mean and variance parameters of each category are obtained to determine a classification function, and then each pixel in the image to be classified is brought into the classification function of each category. The category with the largest return value of the function is regarded as the category of the scanned pixel to achieve the effect of classification.

There are two strategies for selecting separability judgments [61, 62]: (1) select the features with the highest average separability; (2) select the most divisible feature of the most difficult categories. The first strategy is difficult to take care of a more centralized category [63, 64]. If this strategy is used, the choice of balanced care of all types can make up for its shortcomings. Second strategies can take care of the most difficult categories, but it may miss some of the biggest feature

which makes the separability and decreases the classification accuracy [65, 66].

In practical application, the thought of the two strategies should be integrated to achieve balance between efficiency and pattern distribution. If the distribution is more uniform, both strategies should be chosen; but if the pattern distribution is not uniform, to select the first strategy, we must consider the validity of the separability criterion and the most difficult category to improve the classification accuracy.

The aim of this algorithm presented in this paper is to select the features with the highest average separability by using the method of random forest and to distinguish the categories that are easy to distinguish. Then, the weighted entropy algorithm is used to select the most divisible features of the most difficult category. This algorithm of the random forest and the weighted entropy of the optimal parameter combination combined into the classifier proposed in this paper to classify not only can improve the purpose of the classification, but also can improve the accuracy and efficiency of the classification.

By using the traditional minimum distance and support vector machine (SVM) classification method, the AVIRIS and the Hyperion data are used to carry out many experiments on 30% of the total samples and 40% of the classification samples and 50% of the classification samples, which are randomly selected. The experiment is carried out with the classification algorithm proposed in this paper. Among them, there are two methods for sample selection.

Method 1: 5% samples are randomly selected as training samples [66, 67] and the rest samples are used as test data. Then, the performance of the classifier is reflected by the degree of fitting of the classifier and the real ground object [68]. The time of iteration is 1 and the initial number of each feature is 16. Then the most weighted feature category is taken as the category of the feature. Then the weighted entropy algorithm is used to give the category with the highest weighted, which returns the weight of the class predicted by the model [69–75]. 5% additional samples were selected and added to the training sample to form a new training sample,

and then the prediction classification was conducted again, so that the iteration took place until 10 iterations. Among them, 6 iterations, 8 times, and 10 times correspond to 30% of the total number of samples and 40% of the total number of samples and 50% of the total number of samples for many experiments. All of them pick out the total classification accuracy and kappa coefficient of the three corresponding numbers and calculate the corresponding time and average value in order to eliminate the random band. The random training samples corresponding to the minimum distance classifier and the support vector machine classifier are 30%, 40 %, and 50% of the training samples

Method 2 of the experiment: 10% samples were randomly selected as training samples, and the rest were used as test data. The time of iteration is 1 and the initial number of each feature is 16. Then the most weighted feature category is taken as the category of the feature. Then the weighted entropy algorithm is used to give the category with the highest weighted, which returns the weight of the class predicted by the model. 10% additional samples were selected and added to the training sample to form a new training sample, and then the prediction classification was carried out again, so that each iteration took place five times. The number of iterations 3 times 4 times and 5 times corresponds to 30% of the total number of samples and 40% of the total number of samples and 50% of the total number of samples. The mean value of the total classification accuracy, the kappa coefficient, and the corresponding time are calculated to eliminate the errors caused by the randomness, and the random training samples corresponding to the minimum distance classifier and the support vector machine classifier are 30 %, 40 %, and 50 %.

2.1.1. Training Samples Selection and Testing Samples Selection. The purpose of the training samples is to confirm the parameters of the mathematical model. After training, the model system can be regarded as having been established. The purpose of the test samples is to ascertain the function of the model and whether the degree of fitting between the model and real events is small.

In the classification of remote sensing, the selection of the training samples area directly determines the classification results. Using Google Earth to select the training samples area of remote sensing classification is an excellent way and the process is showed in the following:

- (1) Put your research area boundary through ArcGIS tool to KML, the research area in the Google Earth display.
- (2) Sketch the type of object you want on Google and sketch "right-click your folder-save location as", KML file.
- (3) In ArcGIS, convert the KML to layer tool to the ArcGIS-recognized layer, and then the data export the shape file format (note similar dissolved and projection conversions).
- (4) Open your samples vector file in Envi, then export it to the ROI file to the image of you want to classify, and select the attribute for your own defined the type of objects, when it converts to ROI.
- (5) You can see ROI in the main image window you want to categorize.

Training Samples Selection Principle

- (1) Samples distribution should be as wide as possible.
- (2) Choose the pure pixels, not the areas where different features are transferred.
- (3) The relationship between the number of samples and the type of the samples is twice or more.
- (4) The real samples should be consistent with the experimental samples.
- (5) The separability of the training samples is a parameter of reference value to judge the function of the training samples.
- (6) The correlation coefficient within the class should be large, and the correlation coefficient between the classes should be small.

Test Samples Selection Principles

- (1) Band selection should be consistent with the training samples.
- (2) Find the region of interest in other places, which does not coincide with the area of interest of the training samples.
- (3) The real samples should be consistent with the experimental samples.
- (4) Separability is a parameter of reference value to judge the function of test samples.
- (5) The correlation coefficient within a class should be large, and the correlation coefficient between classes should be small.

2.2. Classification Algorithms Selection

2.2.1. The Minimum Distance Classifier. The nearest neighbor method classifies new samples from unknown categories according to a set of samples known to each class of samples, and the classification is based on calculating the distance between the features of the new samples and those of the samples in the set of samples in turn. The nearest neighbor algorithm is mainly based on a limited number of adjacent samples; thus, it is more suited to unclassified data sets with more overlapping parts. Although the nearest neighbor algorithm depends on the limitations of the theorem to some extent, it only needs to consider the adjacent samples' information in the classification, which can solve the problem of sample imbalance. The disadvantage of the neighbor algorithm is that the time complexity of the algorithm is high. The distance between the samples should be classified and each sample in the known samples space should be calculated in turn, and then they will be sorted. In this case, we can process the known samples the first time, remove some samples, and reduce the number of comparisons in the classification process, thus reducing the time consumed by the algorithm [11].

The minimum distance classification is the most basic classification method in the classifier. It is a classification method by calculating the distance between unknown class vector X and the center vector of each previously known class and then reducing the vector X to be classified as the smallest of these distances.

In an n-dimensional space, the minimum distance classification first calculates the mean values of each dimension

of each known class X (expressed as a vector). A mean value is formed, which is represented by a vector (the name of a class, the samples' feature set of category A , the first dimensional feature set of class A , and the mean value n of the first one-dimensional feature set as the total characteristic dimension). The mean value of another category is calculated (expressed as a vector) and used. It is regarded as a samples' feature vector x to be classified. The distance between the two samples is the variable to be calculated. The basic idea of the minimum distance classifier is to generate a central vector representing the class according to the arithmetic average of the training set ($K:1,2,\dots,M$; M is the number of classes), for each data tuple to be categorized X , its distance from the uk is calculated, and finally, it is determined that X belongs to the nearest class.

Here are two values $X=[x_1,x_2,\dots,x_n]$ and $uk=[uk_1,uk_2,\dots,uk_n]$ and c represents category and it belongs to $\{c_1,c_2,\dots,c\}$. Take the Euclidean distance as an example; the formula for calculating the distance is as follows:

$$\begin{aligned} d(x, u_i) &= |x - u_i|^2 = (x - u_i)^T (x - u_i) \\ &= x^T x - (x^T u_i + u_i^T x), \end{aligned} \quad (1)$$

Then look for the minimum value in the two groups. If the former is the smallest, then X belongs to class A , and if the latter is small, then X belongs to class B .

There are many different methods for calculating the distance of classification at present and it is the most common method of calculating distance, Euclidean distance.

Euclidean distance is the most easily understood method of distance calculation, derived from the distance formula between two points in the Euclidean space.

The Euclidean distance between two points of $A(x_1,y_1)$ and $B(x_2,y_2)$ on a two-dimensional plane is as follows:

$$d_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \quad (2)$$

Euclidean distance between two points of $A(x_1,y_1,z_1)$ and $B(x_2,y_2,z_2)$ on two points in three dimensional space is as follows:

$$d_{1,2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \quad (3)$$

Euclidean distance between two points of X_1 (expressed as a vector) and X_2 (expressed as a vector) on two n -dimensional vectors is as follows:

$$d_{1,2} = \sqrt{\sum_{k=1}^n (x_1 - x_2)^2}, \quad (4)$$

The nearest neighbor method classifies new samples in unknown categories according to a set of samples known to each class of samples, and the classification is based on calculating the distance between the features of the new samples and those of the known set of samples in turn. The nearest neighbor algorithm is mainly based on a limited number of adjacent samples, so it is more suited to unclassified data sets

with more overlapping parts. Although the nearest neighbor algorithm depends on the limitation theorem to some extent, it only needs to consider the adjacent samples' information in the classification.

The disadvantage of the neighbor algorithm is that the time complexity of the algorithm is high and the distance between the samples to be classified in the classification process and each sample in the known sample space should be calculated in turn. The nearest samples should be considered after they are sorted. In this case, we can preprocess the known samples' points and remove some samples to reduce the number of comparisons in the classification process, thus reducing the time consumed by the algorithm [11].

2.2.2. Support Vector Machine Classifier. To map the sample space to a high or even infinite dimensional feature space by means of a nonlinear mapping plane, SVM may be a good method. It can transform the nonlinear separable problem in the original sample space into one in the feature space. A linear separable problem involves scaling up and can be linearized. Raising the dimension entails mapping samples to a high-dimensional space. In general, this will increase the computational complexity and even cause a "dimensionality disaster". However, as a matter of classification, regression, and so on, a sample set may not be linearly processed in a low-dimensional sample space. Linear partitioning (or regression) can be realized on a linear hyperplane. The SVM method can solve this problem skillfully by applying the expansion theorem of the kernel function. There is no need to know the explicit expression of nonlinear mapping. Since the linear learning machine is built in the high-dimensional feature space, comparing it with the linear model, the computational complexity is almost not increased. The catastrophe of dimensionality can be avoided to some extent thanks to the expansion of the kernel function and the theory of calculation.

SVM is often used in classification scenarios, and its classification effect is very good. Compared with other classification methods, only when the training samples reach a certain number it can achieve a better result. The algorithm can also obtain satisfactory results on small samples when the number of samples is limited and the classification effect is difficult to guarantee.

The basic principles of the algorithm are as follows: different points and lines represent the classification target and the classification hyperplane, respectively [11]. The optimal hyperplane linear of two classification support vector machines aims to find the optimal hyperplane, so that the two types of distances are maximal and they can be separated correctly. Suppose two classes of samples sets are known: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}, (x_i, y_i) \mid x_i \in R^n, y \in \{-1, +1\}, i = 1 \dots n$

The decision function is

$$f(x) = \text{sgn}(g(x)) \quad (5)$$

The optimal hyperplane description is

$$w \cdot x + b = 0 \quad (6)$$

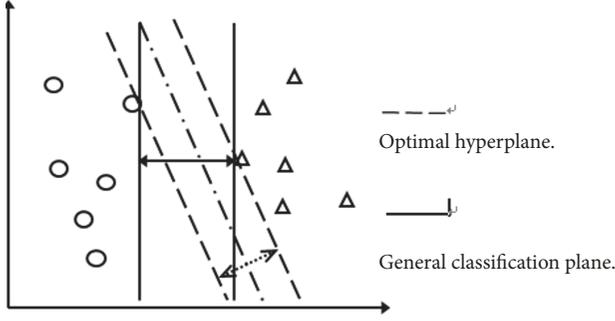


FIGURE 4: The comparison of optimal classification hyperplane and common classification plane.

The result of the classification is

$$\begin{aligned} w \cdot x + b &\geq 0 & y_i &= 1 \\ w \cdot x + b &< 0 & y_i &= -1 \end{aligned} \quad (7)$$

The function w is the normal direction of hyperplane. So as to normalize it, the corresponding relation of classification hyperplane and hyperplane w is established. Therefore, the distance between the nearest samples of judgment surface and judgment surface is $1/\|w\|$, and the interval between the two types is $2/\|w\|$. The samples data set is divided into two areas, i.e.,

$$\begin{aligned} \min_{i=1,2,\dots,n} |(w \cdot x_i) + b| &= 1 \\ y_i [(w \cdot x_i) + b] &= 1 \end{aligned} \quad (8)$$

The classification hyperplane $(w \cdot x) + b = 0$ can classify all samples correctly, that is,

$$y_i [(w \cdot x) + b] \geq 1, \quad i = 1, 2, \dots, n \quad (9)$$

To minimize $\|w\|^2$, the classification surface is the optimal classification hyperplane. The comparison between the general classification surface and the optimal classification hyperplane is shown in Figure 4.

Solving the optimal hyperplane solution of two-class optimal hyperplane can be transformed into two-time programming problems. Constraint condition is

$$\min \Phi(w) = \frac{1}{2} (w \cdot w) \quad (10)$$

$$y_i [(w \cdot x) + b] \geq 1, \quad i = 1, 2, \dots, n \quad (11)$$

Type (11) is a convex programming problem, using the Lagrange multiplier method to solve the upper formula, i.e.,

$$L(w, b, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^n \alpha_i \{y_i [(w \cdot x_i) + b] - 1\} \quad (12)$$

Take the partial derivative of w with respect to b , and set it to zero, and you get

$$\frac{\partial L}{\partial w} = 0 \implies \quad (13)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \implies \quad (14)$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

Formula (13), (14) is brought into (12) and can be

$$W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (15)$$

According to the duality theory of Kuhn-tucker condition, the above problem can be transformed into dual problem, namely,

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (16)$$

Constraint condition is

$$\begin{aligned} \alpha_i &\geq 0, \\ \sum_{i=1}^n \alpha_i y_i &= 0, \\ \alpha_i \{y_i [(w \cdot x_i) + b] - 1\} &= 0 \end{aligned} \quad (17)$$

Solution Type (16): we can get the optimal classification function, namely,

$$\begin{aligned} f(x) &= \text{sgn} \{(w \cdot x) + b\} \\ &= \text{sgn} \left\{ \sum_{i=1}^n \alpha_i^* y_i (x_i \cdot x) + b^* \right\} \end{aligned} \quad (18)$$

In the result, most of α_i are equal to zero, and only the samples x_i responding to the decision boundary distance of 1 are not equal to zero, so the samples α_i corresponding to this part of nonzero α_i are called support vectors. The training samples set will be only a small amount of samples, which can greatly reduce the process of construction and operation, so the efficiency and speed of the support vector machine (SVM) classification method are high.

When the training samples linear inseparable, namely, the training samples cannot be completely separated from the hyperplane of slack variables ξ_i , and error penalty adjustment parameters C can be used to solve the transformation into and the optimal classification function is

$$\min \Phi(w) = \frac{1}{2} (w \cdot w) + C \sum_{i=1}^n \xi_i \quad (19)$$

Constraint condition is

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0, \quad \xi_i \geq 0, \quad i = 1, 2, \dots, n \quad (20)$$

The function of error penalty adjustment parameter C is to control the relationship between the upper bound of samples size and the complexity of the algorithm. The dual problem of transformation is exactly the same as that of the linear separable case but with different constraint conditions, i.e.,

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (21)$$

Constraint condition is

$$\sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n \quad (22)$$

So the final classification function is

$$f(x) = \text{sgn} \{(\mathbf{w} \cdot \mathbf{x}) + b\} \quad (23)$$

An advantage of SVM is that it does not require a lot of samples. This does not mean that the absolute number of training samples is very small, but it is smaller than other training classification algorithms. Under the same problem complexity, the number of samples required by SVM is relatively small, because of the introduction of the kernel function in SVM. Therefore, for high-dimensional samples, SVM is easy to deal with. The structural risk is minimal. This risk refers to the cumulative error between the approximation of the real model of the problem by the classifier and the real solution to the problem. SVM is good at dealing with the inseparability of sample data, mainly through relaxation variables (which are also called penalty variables) and kernel function technology.

2.2.3. The Decision Tree Classifier. The decision tree classification method is a kind of inductive classification algorithm which uses the learning of training samples to excavate the useful rules and use this rule to predict the new Xinji. Its rationale is for each input using a corresponding local model computed from the training data in the region [76]. The basic algorithm of decision tree classification is greedy algorithm, which is a top-down recursive method to construct decision. In each step, it takes the attribute of the best or optimal discrete value field in the current state, and it evaluates the attributes quantitatively by the information gain. The attribute is then established as a standard for partitioning until the data for each node belongs to the same category or no attributes can be used to split the data.

Conventional decision tree rules are generally based on experience and visual interpretation of artificial settings, subject to the influence of subjective factors, and classification and regression tree (Classification And Regression Trees, CART) method can automatically select the classification characteristics and determine the node threshold value. It is the representative of the decision tree model that can handle the nonnumeric data that other algorithms cannot handle [77].

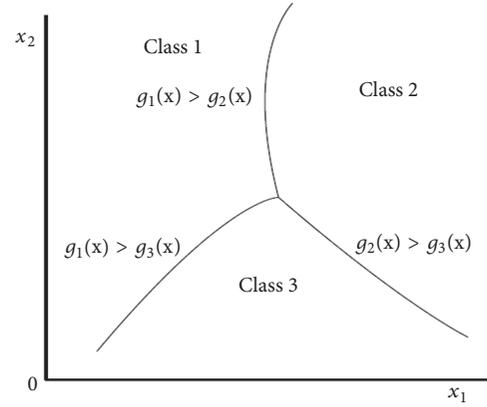


FIGURE 5: Class boundaries by all-at-once formulation.

The basic principle of the CART algorithm is a two-fork tree structure based on the cyclic analysis of the training dataset composed of the test variables and the target variables. CART is a supervised learning algorithm; that is, users must first provide a learning sample set (Learning samples) to build and evaluate the cart before using the cart for forecasting [78]. The variable functions are as follows:

$$L := \{X_1, X_1, \dots, X_m, Y\} \quad (24)$$

$$X_1 := (x_{11}, x_{11}, \dots, x_{1t_1}), \dots, \quad (25)$$

$$X_m := (x_{m1}, x_{m1}, \dots, x_{mt_n}), \quad (26)$$

$$Y := (y_1, y_2, \dots, y_k) \quad (26)$$

$X_1 \sim X_m$ is called attribute vectors, and its properties are continuous and discrete. Y is called a label vector (label vectors) whose properties are contiguous and discrete. When y_i is a continuous quantity value, it becomes a regression number. When y_i is a discrete value, it becomes a classification tree.

Namely, we determine the decision functions:

$$g_i(x) > g_j(x), \quad \text{for } j = i, j = 1, \dots, n. \quad (27)$$

In this formulation we need to determine n decision functions at all once. This results in solving a problem with a larger number of variables than the previous methods.

An example of class boundaries is shown in Figure 5. Unlike one-against-all and pairwise formulations, there is no unclassifiable region.

The way the CART algorithm chooses the split attribute is more interesting. The detailed steps are as follows: first, calculate the impurity and then use the Gini to compute the index. CART chooses a property with the highest information gain; the algorithm uses a greedy top-down approach and each internal node chooses the best classification attribute for the split node. Using the random forest proposed by Breiman is CART in the training process of the decision tree. The decision tree, which is attribute-value-based testing, will enter the training sets, which are divided into subsets, and each of them will be divided into a subset of a repeated

TABLE 1: The process of generate decision tree (using the given training data to produce a decision tree).

Algorithm: Generate decision tree(using the given training data to produce a decision tree)

Input: Training data set samples, with discrete value attributes, the collection of candidate attributes Attribute list;

Output: A decision Tree;

method: (1) Create node N;
 (2) If the samples are in the same Class C;
 (3) returns N as a leaf node, with class C tag;
 (4) If Attribute list is an empty then;
 (5) returns N as the leaf node, marking the most common class in samples;//Majority Voting;
 (6) Selecting the optimal classification attribute test attribute in Attribute list; Using information gain as attribute selection metric;
 (7) The Mark node N is test attribute;
 (8) for the known value AI in each test attribute; The Division of Samples
 (9) It is grown from the node N with a condition of test attribute= ai;
 (10) Set up SI as sample of test attribute=ai in samples; A partition;
 (11) If si is an empty then;
 (12) plus a leaf node, marked as the most common class in the samples;//majority vote
 (13) else plus one by Generate decision tree (SI, Attribute_ List-test attribute) returns the node;

recursively partitioned subset until the next node where all the elements have the same value or attribute value as given or some other stop conditions. We chose optimal segmentation nodes aimed at dividing the data set into homogeneous subsets as far as possible. Because entropy expresses the content of information, the smaller the entropy value the more ordered the subset, and a bigger entropy Gini means better homogeneity of the subsets.

Gini impurity is the expected error rate at which a certain result from a set is randomly applied to a data item in the set. It can be calculated as the sum of the product of each selected probability and the probability of this misdivision. If all data on the point belong to a certain target class, then the Gini impurity gets its minimum value of 0.

The classification algorithm of this paper uses random forest with probabilistic output to calculate the probability of each pixel in hyperspectral remote sensing image. The random forest using in this paper with probabilistic output is based on CART (Classification and Regression Tree) decision tree algorithm and the detailed steps of CART is showed at Table 1.

The processes of decision tree classification are as follows:

(1) The establishment of decision tree model; (2) decision tree classification in ENVI; (3) accuracy evaluation in ENVI. The steps are showed as in Figures 6, 7, and 8.

The advantages of CART: the decision tree classification method has the characteristics of clear structure, repeatable operation, high efficiency, flexibility, and intuition and has a good effect in remote sensing image classification. In the decision tree algorithm, there are C5.0 algorithm and classification regression tree CART (classification and regression trees) algorithm, and the classification accuracy of CART decision tree algorithm is better than that of C5.0 algorithm and has the advantages of clear structure and so on. Conventional decision tree rules are generally based on experience and visual interpretation of artificial settings,

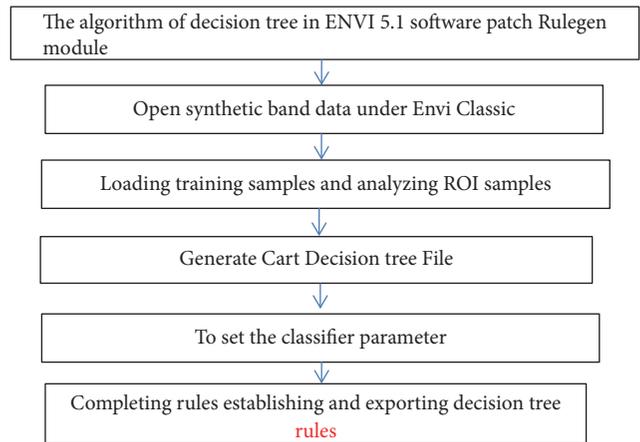


FIGURE 6: Flowchart of the establishment of decision tree model.

subject to the influence of subjective factors, and classification and regression tree (classification and regression Trees, CART) method can automatically select the classification characteristics and determine the node threshold value [79]. It is the representative of the decision tree model to deal with the nonnumeric data which other algorithms cannot handle.

2.2.4. Semisupervised Classification of Random Forest Based on Weighted Entropy. The process of random forest classification involves classifying each randomly generated decision tree classifier, input feature vector, and forest tree to classify the samples. Based on the weight of each tree, the final classification result is obtained. All tree training instances use the same parameters and a different training set; the error estimation of classifiers is based on out of bag. The method of bagging is used to generate different training sets. In other words, bootstrap sampling is used to generate new training sets from the original training set. For each new training set,

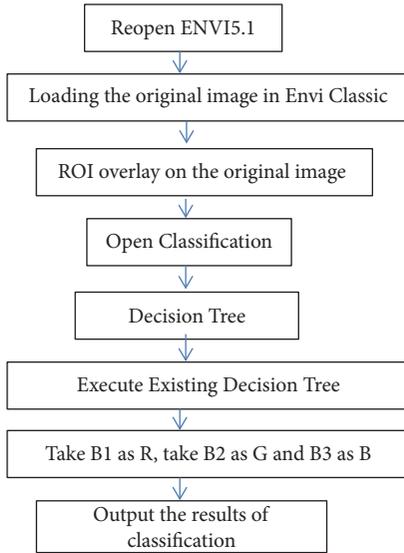


FIGURE 7: Flowchart of decision tree classification in Envi.

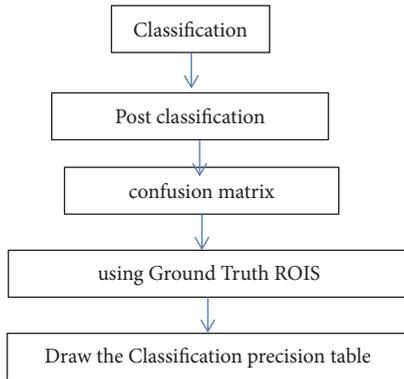


FIGURE 8: Flowchart of accuracy evaluation.

the random feature selection method is used to generate the decision tree, and the decision tree is not pruned during the growth process.

Random forest is a bagging integrated classifier with CART decision trees as weak classifiers. Boosting is usually used to iteratively call the weak classifier learning algorithm to construct a series of weak classifiers. Subsequently, each round gives greater weight to the failed samples of the last round. Bootstrap aggregating is improved by combining randomly generated training sets to select the training samples independently from the same distribution. There are many algorithms of decision trees for each weak classifier, such as iterative dichotomy (ID3), decision tree category methods (C4.5), and CART.

The core of the ID3 algorithm is the application of the information gain criterion to the decision tree nodes. The method of constructing the decision tree recursively is as follows: start from the root node, calculate the information gain of all possible features of the node and select the maximum characteristic of the information gain as the characteristic of the node, and establish the child nodes based on the different

values of the feature. Then, recursively call the above method to the child node and construct the decision tree. Repeat the same steps until all characteristics of the information gain are very small or no characteristics can be selected. Finally, obtain a decision tree. The disadvantages of ID3 are that (1) when we select attributes with information gain, we prefer to select the attribute value with many values, that is, the attribute with more value; and (2) it cannot process contiguous properties.

The C4.5 algorithm is one of the top 10 algorithms in data mining. It is an improvement on the ID3 algorithm, with the following modifications:

- (1) it can use the information gain ratio to select attributes;
- (2) it can prune trees during the construction of decision trees;
- (3) it can be processed for nondiscrete data; and
- (4) it can handle incomplete data.

Because random forests are composed of a series of CART (classification and regression tree) decision trees, and vote by the decision tree, the attribute metric used is Gino index [67]. Suppose the information source x is a discrete random variable, and the value of X is $X = \{x_1, x_2, \dots, x_n\}$. If the probability of each message happening is $P = \{p_1, p_2, \dots, p_n\}$, in addition to $\sum_{i=1}^n p_i = 1$, Gini index concrete formula is as follows:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (28)$$

The smaller the probability of a category appearing in D means the smaller the Gino index value and the higher the "purity" of the sample. For attributes A in the training sample data set D that will be divided into $D1$ and $D2$, the following formula shows the Gini index for the given division of D :

$$Gini_A(D) = \left| \frac{D1}{D} \right| Gini(D1) + \left| \frac{D2}{D} \right| Gini(D2) \quad (29)$$

For discrete value attributes, recursive selection of this attribute produces a subset of the smallest Gini index as its split subset.

For continuous value attributes, all possible split points must be considered, and the decision is similar to the information Gini processing method introduced in ID3; its formula is as follows:

$$Gini_A = \sum_{i=1}^v \left| \frac{D_i}{D} \right| \times Gini(D_i) \quad (30)$$

The point where the given continuous attribute value produces the smallest Gini index is chosen as the split point of the attribute, which is the number of nodes [35]. When CART (classification and regression tree) is constructed, each node t is marked with a corresponding class, regardless of whether the nodes in the decision tree are divided or not. Inequality is used as the criterion of classification:

$$\frac{P_C(j|i) \cdot P(i) \cdot Ni_t}{P_C(i|j) \cdot P(j) \cdot Nj_t} > \frac{Ni}{Nj} \quad (31)$$

If all classes except node I are true, then node t is marked as class, where the a priori probability of class is represented by the a priori probability of class, which is the number of

classes in the sample of node t and the cost of dividing node t into classes. This can be found by looking up the decision tree matrix.

The probability formula x of a result variable of semisupervised classification random forest model with probability output represents a category set; and c solving formula is as shown in the following formula:

$$\hat{c} = \arg \max (c | x), \quad c \in (1, \dots, N_C) \quad (32)$$

A semisupervised hyperspectral remote sensing image classification method based on weighted entropy is characterized in that the categories corresponding to the probabilistic output of each pixel category in the function (33) are

$$CLA(x_i) = \arg \max \sum_{i=1}^c P \quad (33)$$

In the function of (29), the semisupervised random forest hyperspectral remote sensing image classification method based on weighted entropy outputs the classification results and evaluates the accuracy. If the first iteration is suitable for the results, the subsequent steps are carried out; otherwise, the results are compared with the previous output results. If the difference between the two is greater than the given threshold, the subsequent steps are conducted. If the difference is less than the given threshold, the final result is output.

The purpose of this paper is to transform the uncertainty label into a deterministic label that is expressed as an entropy value. In weighted implementation, the greater entropy is given the greater weight. This helps to distinguish the difficult surface categories. It regards the large entropy value as the key distinguishing object that can improve the efficiency of the classification.

In the steps of the semisupervised random forest hyperspectral remote sensing image classification method based on weighted entropy, the weighted entropy algorithm based on voting probability is used to assign different weights to ground objects according to the different needs of researchers [17]. The probability of each pixel in the remote sensing image is transformed into the uncertainty formula by converting the probability into uncertainty, as shown by the following formula:

$$W(X) = - \sum_{i=1}^q w_i p_i \log p_i \quad (34)$$

Suppose the information source x is a discrete random variable and the value of X is $X = \{x_1, x_2, \dots, x_n\}$. If the probability of each message happening is $P = \{p_1, p_2, \dots, p_n\}$, in addition to $\sum_{i=1}^n p_i = 1$, the function of the weighted entropy algorithm takes into account the degree of attention to information and the influence of events on people. When we calculate the weighted entropy value of pixel of hyperspectral image data, the maximum value will occur. In order to ensure the accuracy of the experimental results, a normalized treatment is adopted, as shown by the following formula:

$$R(X) = - p_i \sum_{i=1}^q w_i p_i \log p_i \quad (35)$$

We use the OA (the overall accuracy) and kappa to show the function of the classification, which are showed at formulae (36) and (37). OA is the ratio of the number of validation pixels that have been correctly classified to the total number of validation pixels used for all classes and is expressed as a percentage (%). Kappa is the proportion of correctly classified validation points after random agreements are removed and it expresses the extent to which the confusion matrix results are not obtained by chance or random, which is showed at the following formula:

$$OA = \frac{\sum_{i=1}^m tp_i}{n} \quad (36)$$

$$Kappa = \frac{n \sum_{i=1}^m (tp_i) - \sum_{i=1}^m ((tp_i + fp_i) \times (tp_i + fn_i))}{n^2 - \sum_{i=1}^m ((tp_i + fp_i) \times (tp_i + fn_i))} \quad (37)$$

In the function, fp_i is the main diagonal element in i_{th} row, which is the ratio of the number of validation pixels that have been correctly classified; fp_i is computed from the sum of i_{th} row column which is the number of real pixels in a class, excluding the main diagonal element; fn_i is the sum along i_{th} row, which is the number of classified pixels in this class, excluding the main diagonal element; m is the number of classes; and n is the total number of pixels in all surface real categories.

2.3. The Processing of the Algorithm Proposed in the Paper.

The purpose of this improved algorithm is to select the most difficult category and find out the most divisible feature and then classify it and improve the accuracy of classification. The training convergence rate is slow and the performance of each classification varies greatly. The detailed steps of the algorithm are showed at Table 2.

Figure 9 shows the main technical flowcharts of this experiment, which is presented as in Figure 9.

3. Results

This experiment is based on the following computer hardware devices:

Inte: (R) Core(TM) i5 3230M CPU @ 2.6Ghz 2.6 Ghz
Install RAM Ram: 4.00GB
System type: 64-bit operating system
The software environment is as follows:

In the environment of Microsoft Windows 7, Envi/IDL5.2 and MATLAB 2017b are used to carry out the experiment.

In this experiment, AVIRIS and Hyperion data were used, respectively.

There will be two parts in this chapter. First of all, the discussion of parameters selection the parameters of random forest will be illustrated. In the second part, classification results and analysis will be showed.

3.1. The Discussion of Parameters Selection. In order to ensure the random forest classifier performs well, the number of variables used in a random forest decision tree (N-tree) and

TABLE 2: The process of semisupervised classification of random forest based on weighted entropy algorithm.

Algorithm: Semi-supervised Classification of Random Forest Based on Weighted Entropy

Input: Hyper-spectral remote sensing image data; training sample set; the category set corresponding to the training sample.

Output: The results of classification images and confusion matrix;

method:

- (1) Random forest with probability output is used to determine the expected value of the category of ground objects with the largest number of votes;
- (2) Judging the type of the samples according to the results of output;
- (3) Judging the accuracy of the output results according to the classified ground object classification data;
- (4) Then the weighted entropy algorithm is used to give the category with the highest weighted return weight of the class predicted by the model;
- (5) The ground objects which account for 5% or 10% of the total sample increase are selected and added to the training sample to form a new training sample, and then the prediction classification is carried out again;
- (6) The unlabeled label pixel in the input hyper-spectral image is converted into the tag pixel according to the uncertainty evaluation value;
- (7) A new tag is added to the original training set and a new training set is constructed;
- (8) Run iteratively until termination requirements are met or unlabeled training samples run out.
- (9) Using the remaining samples to test the performance of the classifier, that is, to evaluate the accuracy of the classifier;
- (10) In order to test the classification performance and the universality of the classifier, the hyper-spectral remote sensing images of Hyperion and the hyper-spectral remote sensing images of AVIRIS type are classified, and the accuracy is evaluated.

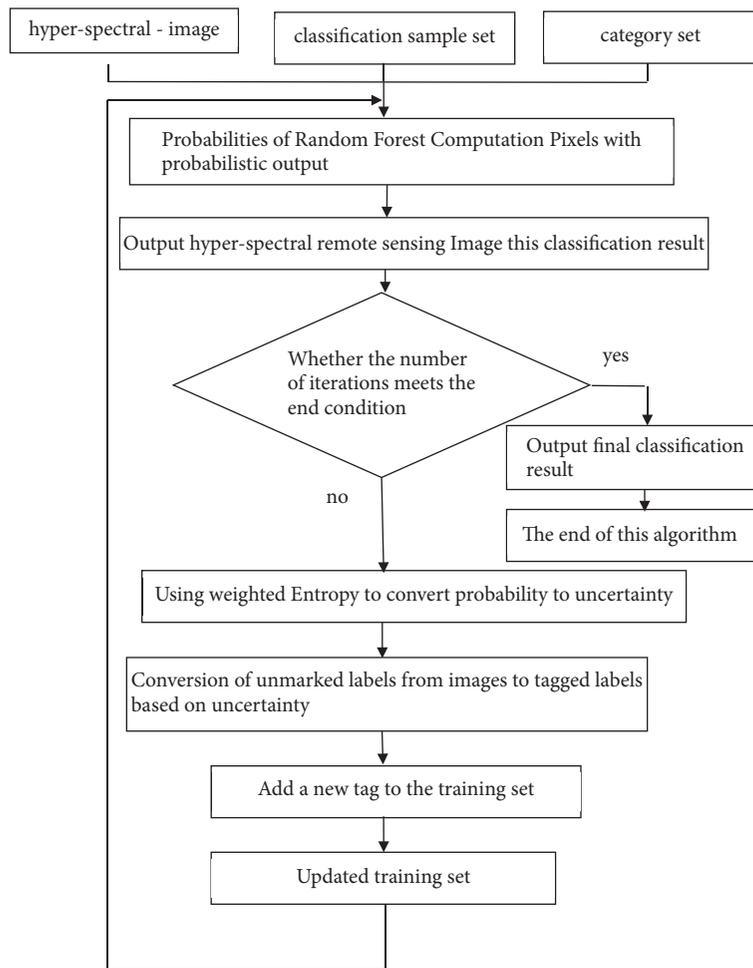


FIGURE 9: Flowchart of classification framework.

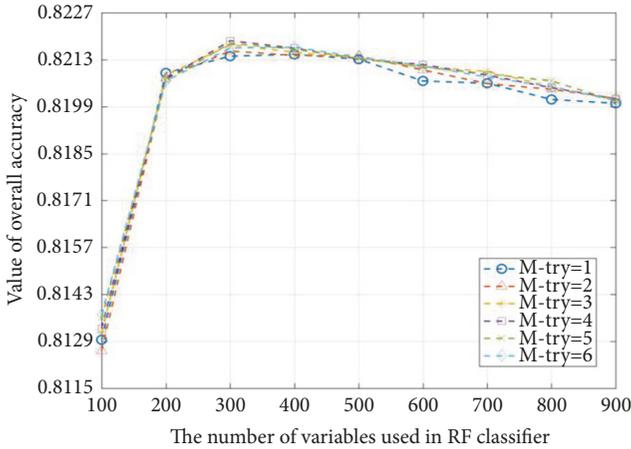


FIGURE 10: The tendencies of overall accuracy when the N_tree and M_try are setting different values for AVIRIS.

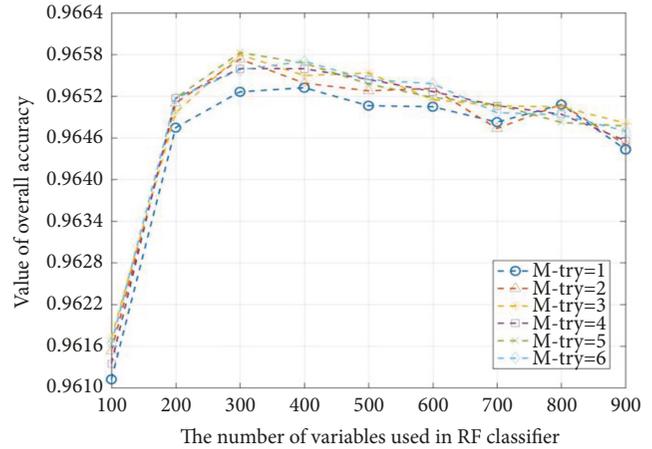


FIGURE 12: The tendencies of overall accuracy when the N_tree and M_try are setting different values for Hyperion.

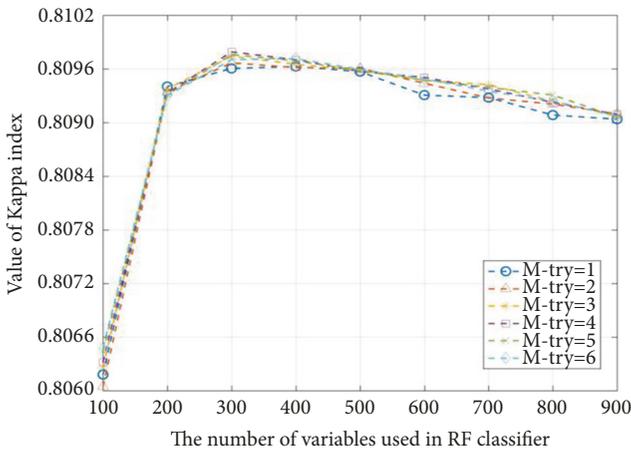


FIGURE 11: The tendencies of kappa when the N_tree and M_try are setting different values for AVIRIS.

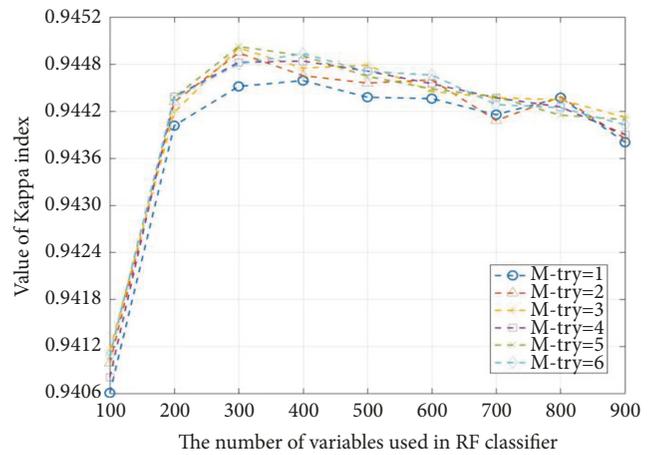


FIGURE 13: The tendencies of overall accuracy when the N_tree and M_try are setting different values for Hyperion.

the node number (M-try) should be selected. Among them, the characteristic number M-try decides to construct the correlation between the ability of the decision tree and the decision tree; the number of decision trees (N-tree) determines the number of votes and the accuracy of the random forests. Owing to the limitations of real conditions, the N-tree used in random forest changes from 100 to 1000, and M-try changes from 1 to 9. 90 when experiments are performed, aiming at selecting the most appropriate parameters for every data source used in the experiments.

The optimal combination of the number of decision trees and the number of nodes is selected. After testing 90 times, a set of optimal combinational parameters that are suitable for random forests are obtained; these are shown in Figures 10–13. Here, the most fitting parameters for random forest are selected. To be of scientific merit, the evaluation parameters must show the classification function of the classifiers, and their high values show that the fitting degree is good. The figures show that when the combination parameter in that decision tree's selection is 300 and the node number selection

is 4, it makes the evaluation parameters of the results of the classification, and it has a high value for AVIRIS and Hyperion data. Therefore, we selected combination parameters where N-tree was 300 and M-try was 4 to carry out the random forest model before starting the weighted entropy algorithm, which ensured the semisupervised classification ran well.

3.2. Classification Results and Analysis. Conducting a qualitative and quantitative analysis of the classification results in this part aiming at evaluating the accuracy of the classification.

3.2.1. The Conduct of Qualitative Analysis of the Results of Classification. While employing the traditional minimum distance and SVM classification method, the AVIRIS and Hyperion data were used to carry out many experiments on 30% of the total samples, 40% of the classification samples, and 50% of the classification samples, which were randomly selected. The mean value of the total classification accuracy,

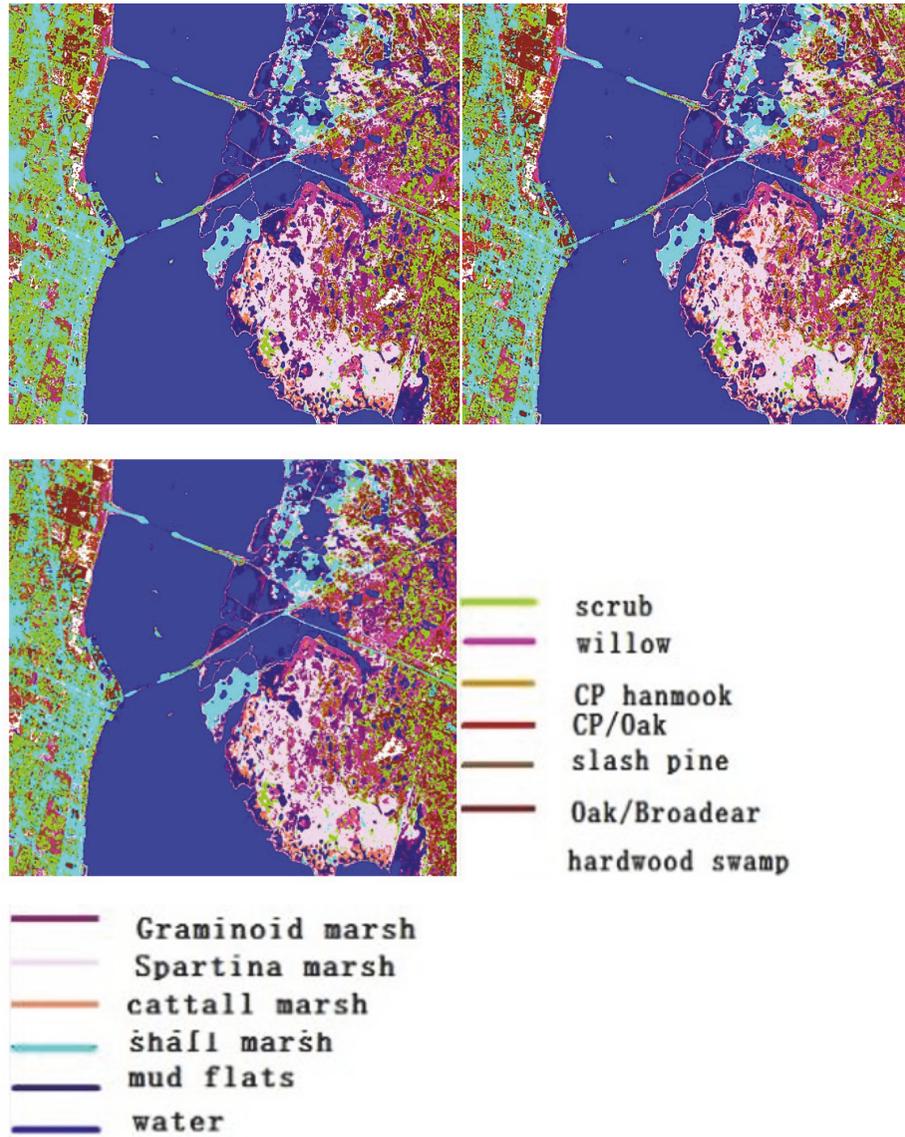


FIGURE 14: The classification results of minimum distance algorithm for AVIRIS data.

the kappa coefficient, and the corresponding time were calculated so as to eliminate the errors caused by the randomness, and the random training samples corresponding to the minimum distance classifier and the support vector machine classifier were 30%, 40%, and 50%. The classification results are showed from Figures 14–24.

From the map of the classification results, we can see that the classification results of the algorithm proposed in this paper were very consistent with the real surface situation, but showing the classification results more vividly is a problem. In order to show the results more vividly, a qualitative method is proposed, which is illustrated in the next part.

3.2.2. The Conduct Quantitative Analysis of the Results of Classification. The performance of the classifier can be demonstrated more intuitively according to the overall classification accuracy, kappa coefficient, and running time. Under

a condition where the initial label selects 5% and 10% of the total samples, the total precision of the semisupervised classification method proposed in this chapter is shown in Figures 25–32 for 10 iterations and 5 iterations. The AVIRIS data for method 1 and method 2 are from Figures 25–28. The Hyperion data for method 1 and method 2 are from Figures 29–32.

Figure 25 clearly shows the trends in classification accuracy under different initial conditions and iterative states of the AVIRIS data. It can be seen from the diagram that the number of starting samples was selected as 5% of the total samples; that is, 16 ground objects of each sample were selected for the experiments on this algorithm.

The classification accuracy of the corresponding iteration was 80.02, and then the samples with larger entropy values were added in order from large to small, and 5% of the total number of samples were selected as the new training samples

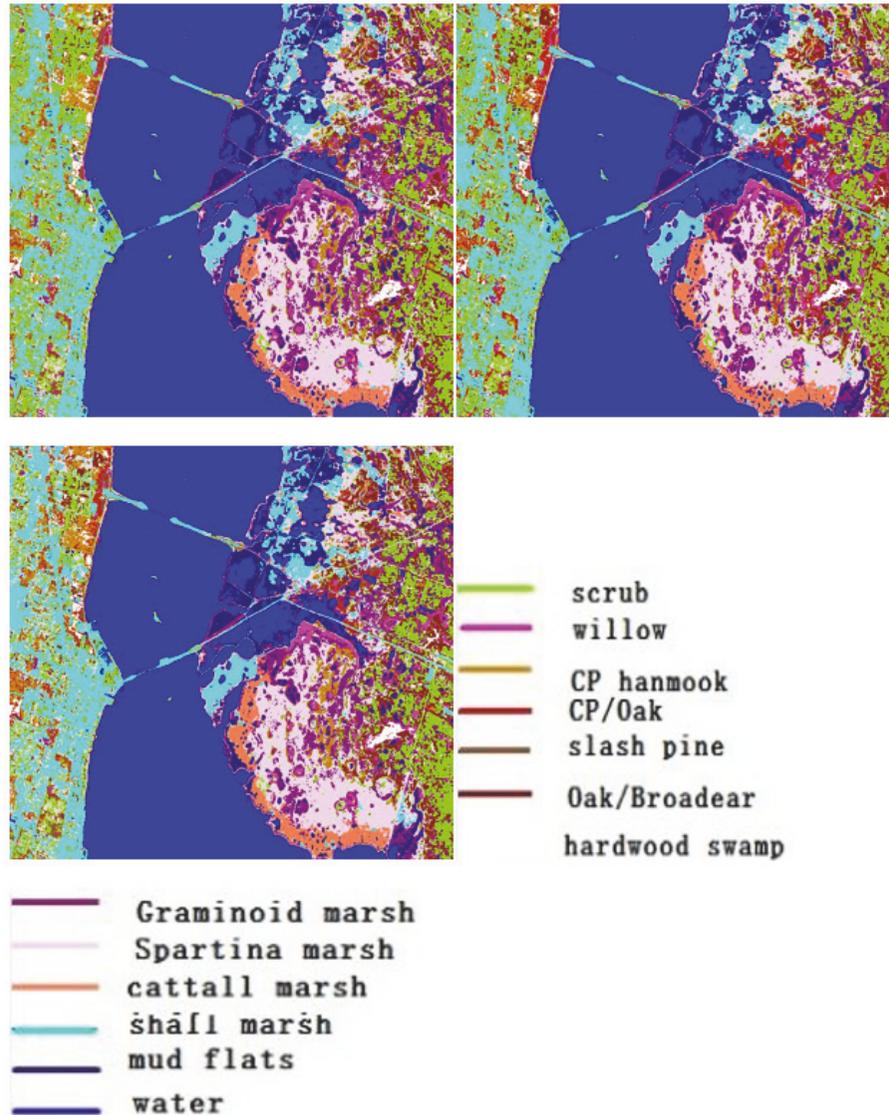


FIGURE 15: The classification results of support vector machine algorithm for AVIRIS data.

for the second iteration. Using the above rules, we found that, with the increase in the number of samples, the classification accuracy was obviously improved. After the 10th iteration, the kappa reached 0.8542.

The second group of experiments was intended to test the other steps when the initial sample was set at 10% of the total training samples. From Figure 26, we can see that the initial classification accuracy was 83.82, which means the number of training samples increased with the number of iterations. The overall precision increased rapidly and the precision curve tended to be gentle after the fourth iteration; the precision was 87.36 after the fifth iteration. Due to the control of the variables in the two experiments, we found that the classification accuracies of the different initial sample numbers were different: the classification accuracy was higher when the initial sample number was larger.

Compared with the same group of experiments, it was found that the overall accuracy of the classification obviously increased with the increase in iteration time, which indicates that the classification accuracy of the algorithm proposed in this paper has a great correlation with the growth of the samples set.

A comprehensive analysis of the classification results on the AVIRIS data showed that the algorithm proposed in this paper functions relatively well and the initial sample number and the growth of the samples set are closely related to the classification accuracy; also, the initial tag number corresponding to higher classification accuracy is significantly increased with an increase in training samples. What is more, when the samples set is increased to a certain point, the overall accuracy increase is no longer obvious.

Figure 27 clearly shows the trends in the kappa different initial and iterative states of the AVIRIS data. From the

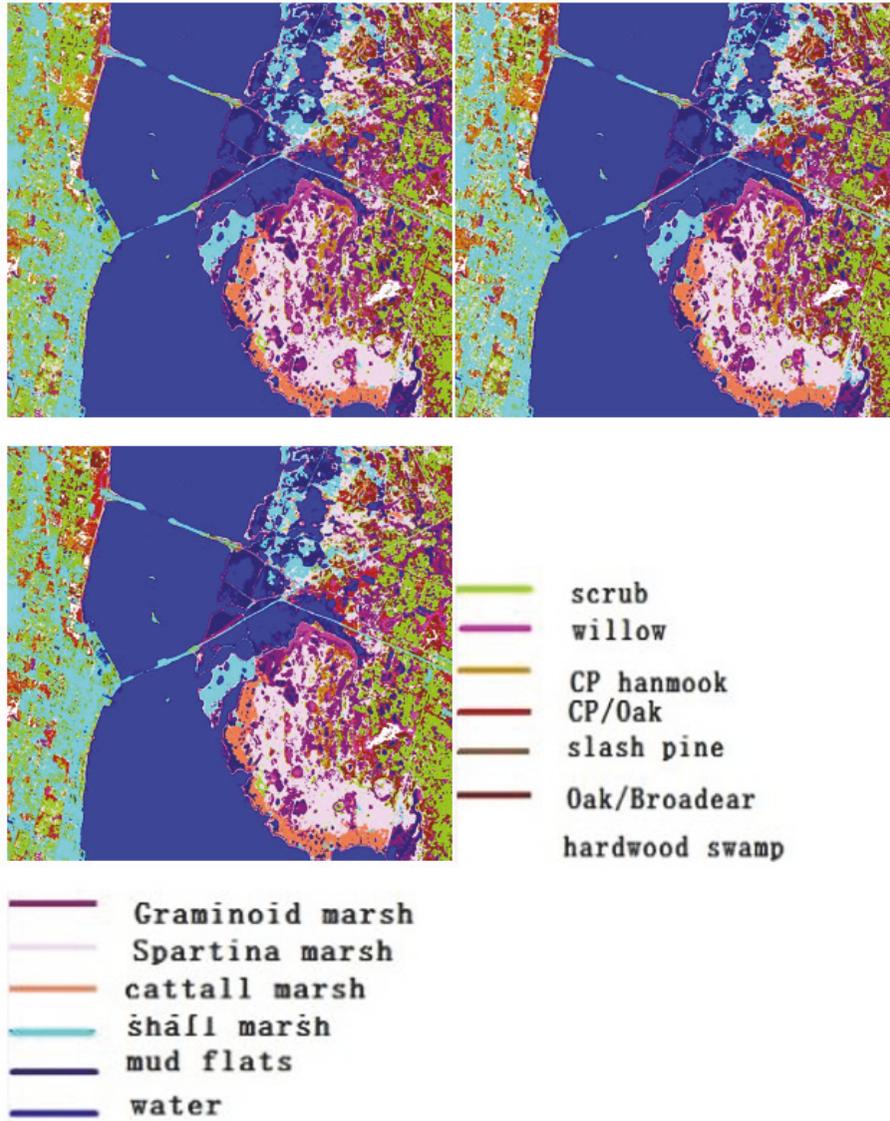


FIGURE 16: The classification results of CART algorithm for AVIRIS data.

diagram, we can see that the starting samples were 5% of the total samples; that is, 16 ground objects from each sample were selected for the experiment on this algorithm.

The corresponding kappa was 0.7798 at the first iteration, and then we added the larger samples sorted according to entropy value from large to small and selected 5% of the total samples as the new training samples for the second iteration to carry on the experiment. In accordance with the above rule, the kappa coefficient was obviously raised with the increase in the sample number, until the 10th iteration. After the 10th iteration, the kappa reached 0.8371.

The second group of experiments was intended to test the other steps when the initial sample was set as 10% of the total training samples. As shown in Figure 28, the initial kappa was 0.8199 with the increase in the number of iterations. What is more, the kappa coefficient increased rapidly with the increase in the number of training samples. After the fourth

iteration, the trend of the kappa curve became stable and the kappa was 0.8591 after the fifth iteration. Also, compared with the experimental group, we found that kappa increased significantly with the increase in the number of iterations, which indicates that the kappa algorithm proposed in this paper is strongly related to the growth of samples set.

A comprehensive analysis showed that the algorithm proposed in this paper on the AVIRIS spectrometer data acquired the classification results and the initial number of samples and the samples set were closely related to the growth of the number of initial labels; also, the corresponding kappa was higher. At the same time, the corresponding kappa significantly increased with the increase in training samples. When the samples set increased to a certain point, the kappa increase was no longer so obvious.

Figure 29 clearly shows the trends in classification accuracy in different initial and iterative states of the Hyperion

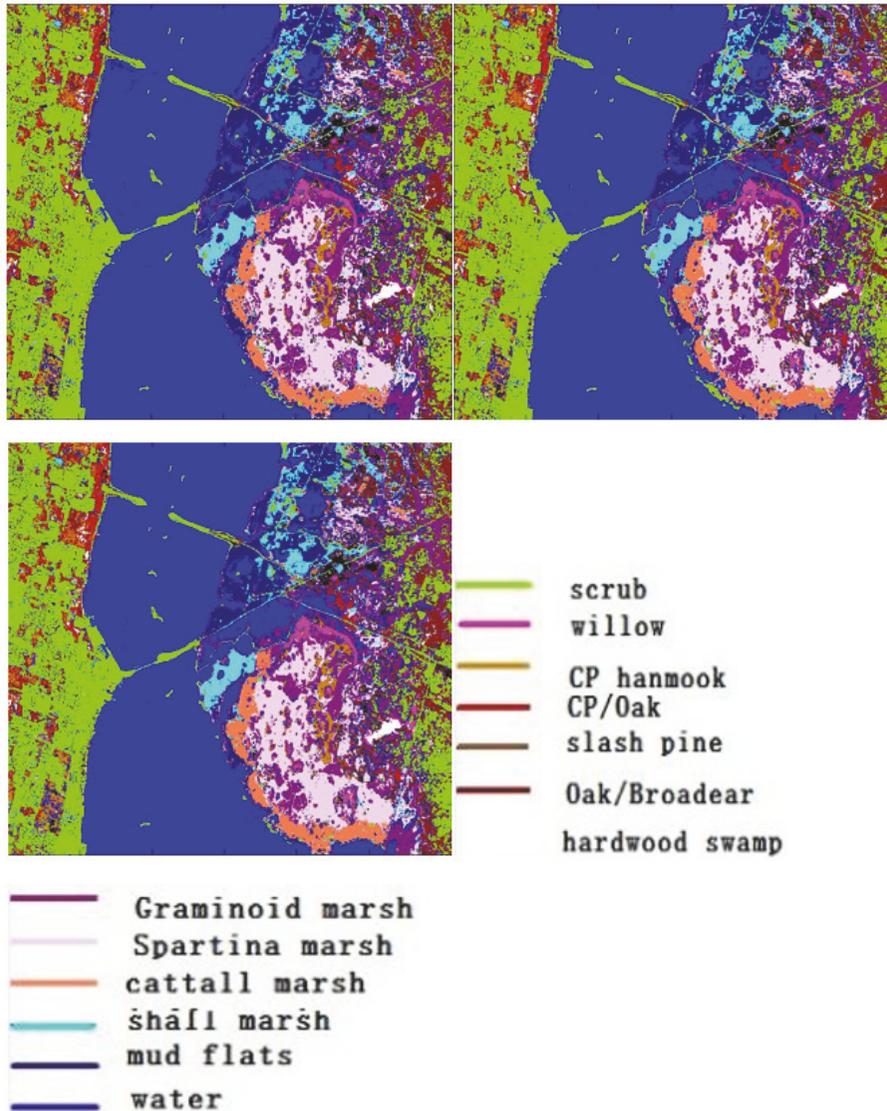


FIGURE 17: The classification results in the initial 5% label conditions of the algorithm for AVIRIS data.

data. It can be seen from the diagram that the starting sample was set as 5% of the total samples; that is, 16 ground objects from each sample were selected for the experiment on this algorithm. The total accuracy of the corresponding iteration was 0.9352, and then the samples with larger entropy values were added in order from large to small, and 5% of the total samples were selected as the new training samples for the second iteration. Using the above rules, the classification accuracy was obviously improved with the increase in the sample number up to the 10th iteration.

In the second group of experiments, the initial samples were set at 10% of the total. Looking at Figure 30, we can see that the initial classification accuracy was 98.20; that is, the number of training samples increased with the number of iterations. The overall precision increased rapidly, and the precision curve tended to be gentle after the fourth iteration and 99.17 after the fifth iteration. It was found that the

classification accuracy differed based on the initial sample number: more initial samples meant more classification precision in the same conditions. The higher the degree of classification means the higher the classification accuracy, and the higher the number of iterations means the higher the classification accuracy, which indicates that the classification accuracy of the proposed algorithm is highly correlated with an increase in the sample set.

A comprehensive analysis showed that the classification effect of the proposed algorithm meant higher classification accuracy. The higher the number of training samples the higher the classification accuracy, and the higher the initial tag number the higher the classification accuracy. However, when the sample set was increased to a certain point, the increase in the total accuracy was no longer obvious.

Figure 31 shows the kappa trends in different initial and iterative states of the Hyperion data. From the diagram, we

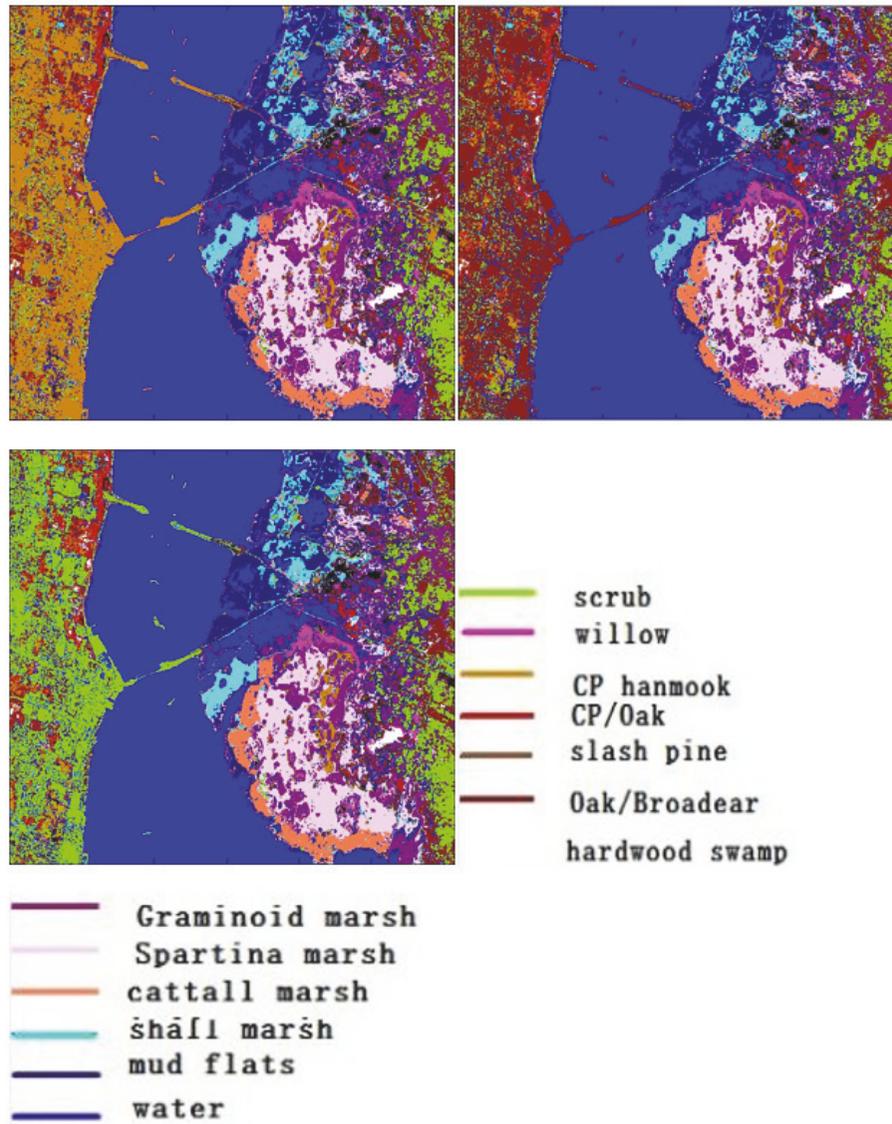


FIGURE 18: The classification results in the initial 10% label conditions of the algorithm for AVIRIS data.

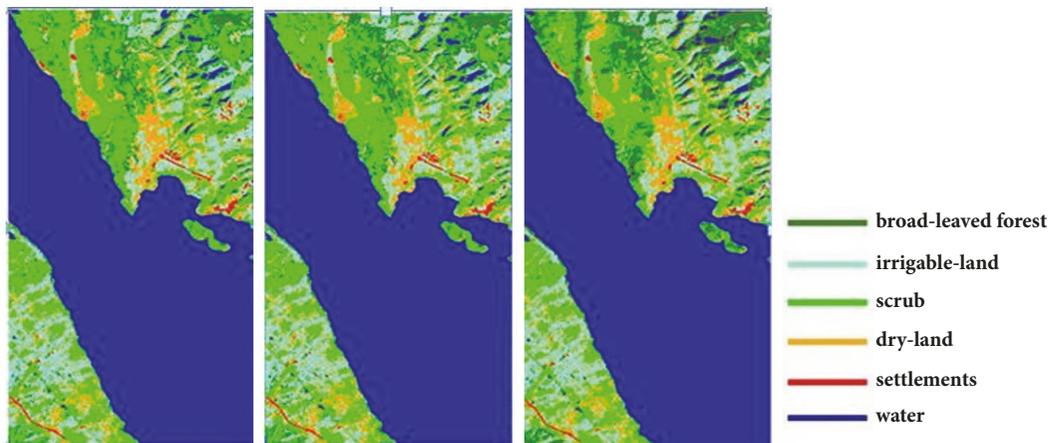


FIGURE 19: The classification results of minimum distance algorithm for the Hyperion data.

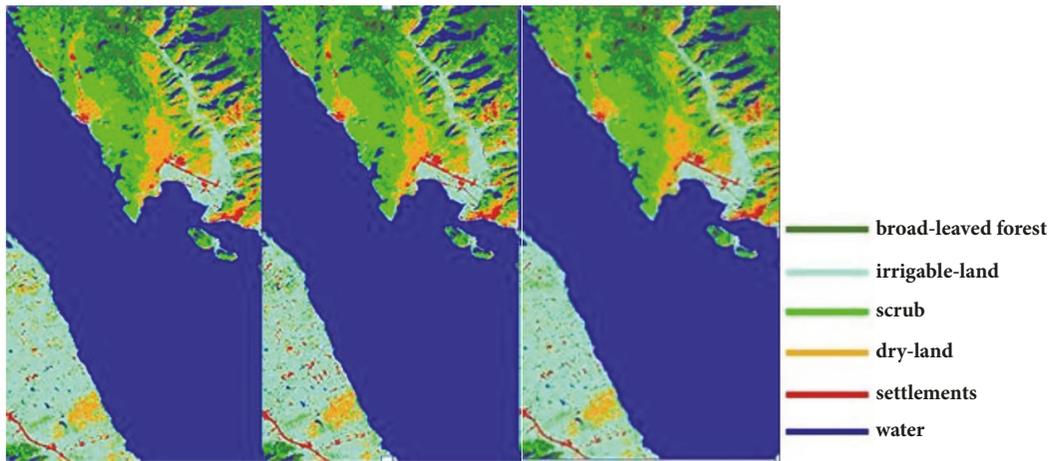


FIGURE 20: The classification results of support vector machine algorithm for the Hyperion data.

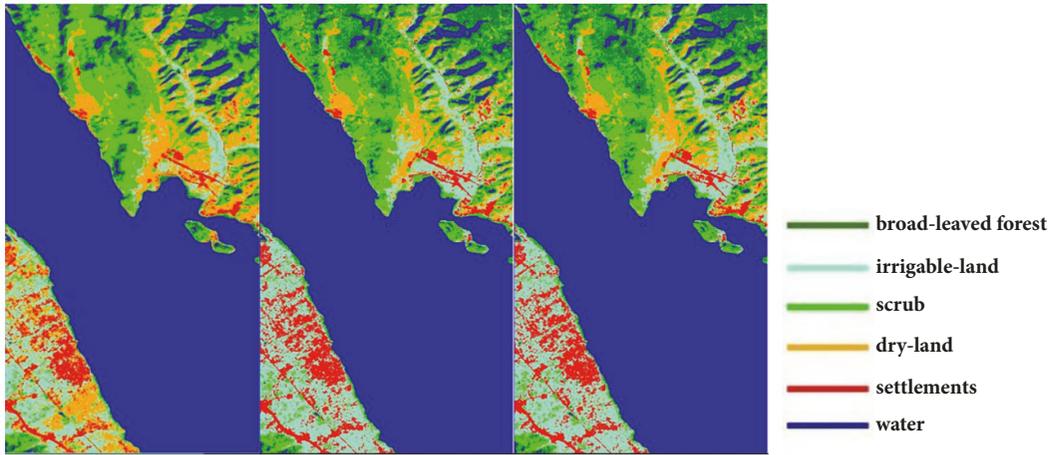


FIGURE 21: The classification results of support vector machine algorithm for the Hyperion data.

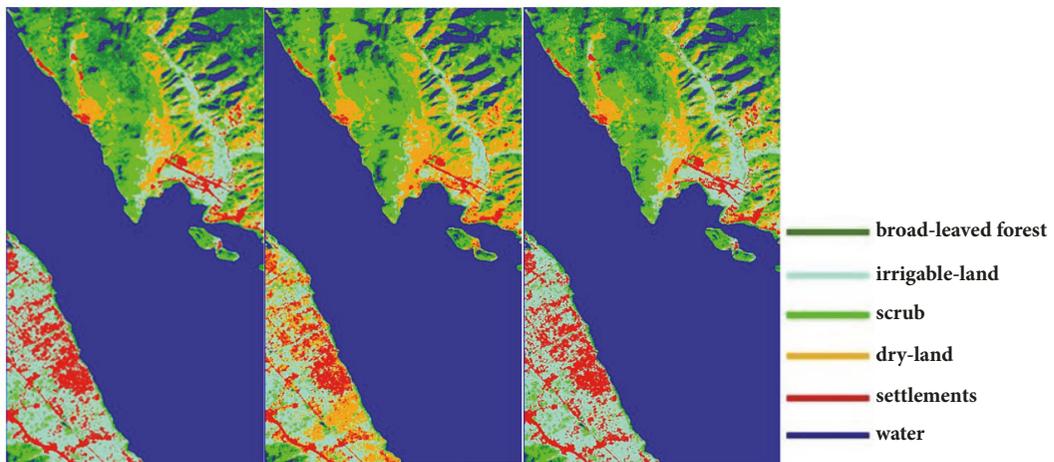


FIGURE 22: The classification results CART algorithm for the Hyperion data.

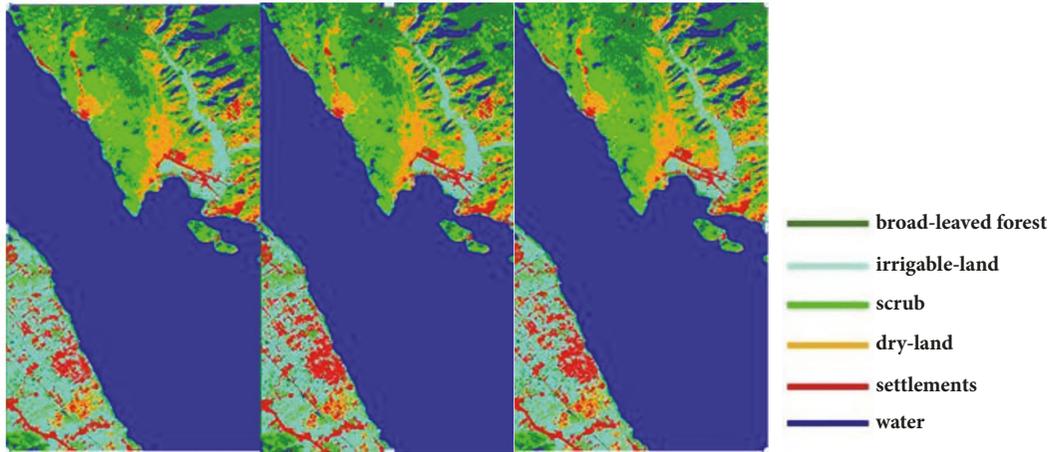


FIGURE 23: The classification results in the initial 5% label conditions of the algorithm for the Hyperion data.

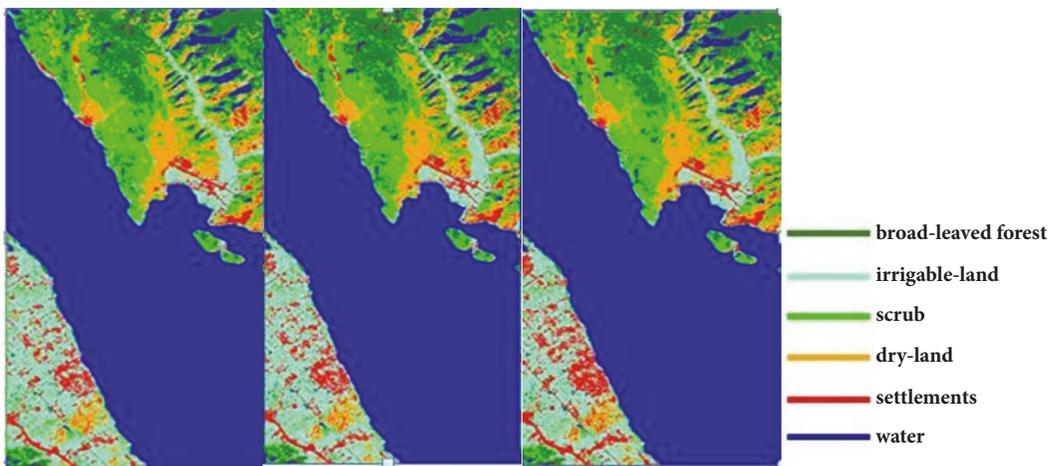


FIGURE 24: The classification results in the initial 10% label conditions of the algorithm for the Hyperion data.

can see that the starting samples were set at 5% of the total samples; that is, 16 ground objects from each sample were selected for the experiment on this algorithm. The kappa of the corresponding iteration was 0.8812. Then, the samples with larger entropy values were added in order from large to smaller, and 5% of the total samples were selected as the new training samples for the second iteration. Using the above rules, we found that the kappa coefficient increased with the number of samples until the 10th iteration. After the 10th iteration, the kappa reached 0.9888.

In the second group of experiments, the number of initial samples was set at 10% of the training samples. As shown in Figure 32, the initial kappa was 0.9255, and as the number of iterations increased, the final kappa reached 0.9901. The kappa coefficient increased rapidly when the number of training samples increased. By the fourth iteration, the kappa curve tended to be smooth, and its value was 0.9904. When the number of initial samples was higher, the kappa was different. Compared with the same group of experiments, it was found that the number of iterations obviously increased,

which indicates that the kappa of the proposed algorithm has a greater correlation with the increase in the samples set.

Our analysis showed that the classification effect of the proposed algorithm for the Hyperion data was closely related to the initial sample number and the growth of the samples set. To some degree, the higher the initial tag numbers the higher the corresponding kappa. However, once the sample set was increased to a certain point, the increase in kappa was no longer so obvious.

The classification performance of the minimum distance classification algorithm, the SVM classifier, and the semisupervised classifier based on weighted entropy and stochastic random forest integration can be classified using the same validation data. The results on the overall classification accuracy, the kappa coefficient, and the running time are shown in Tables 3 and 4.

The tables show that when they are in the same conditions, the semisupervised classifier based on weighted entropy and the stochastic random forest integration proposed in the paper function well, which improves the overall

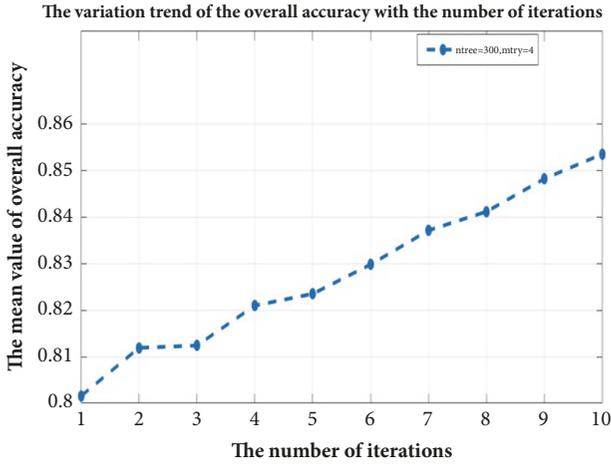


FIGURE 25: Classification accuracy trend of different initial and iterative condition of AVIRIS data for method 1.

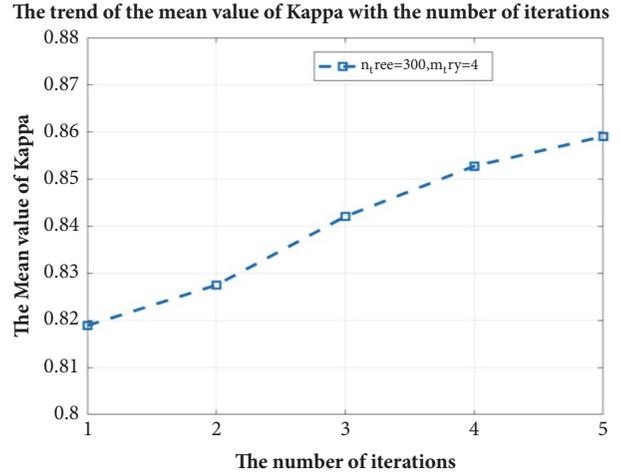


FIGURE 28: Classification kappa trend of different initial and iterative condition of AVIRIS data for method 2.

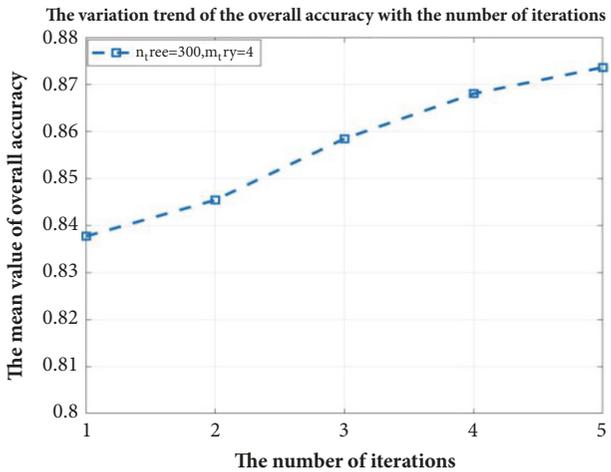


FIGURE 26: Classification accuracy trend of different initial and iterative condition of AVIRIS data for method 2.

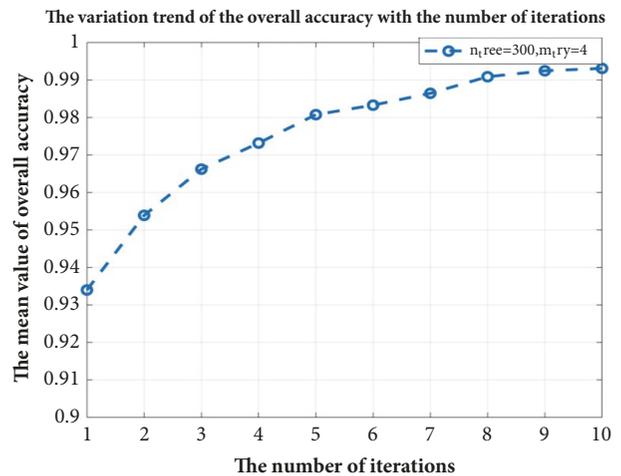


FIGURE 29: Classification accuracy trend of different initial and iterative condition of Hyperion data for method 1.

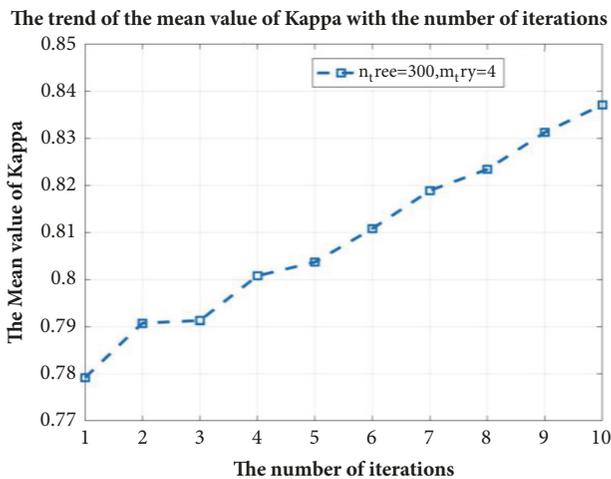


FIGURE 27: Classification kappa trend of different initial and iterative condition of AVIRIS data for method 1.

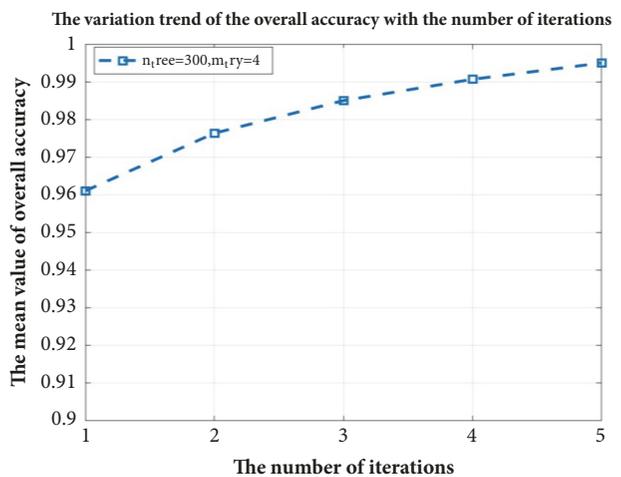


FIGURE 30: Classification accuracy trend of different initial and iterative condition of Hyperion data for method 2.

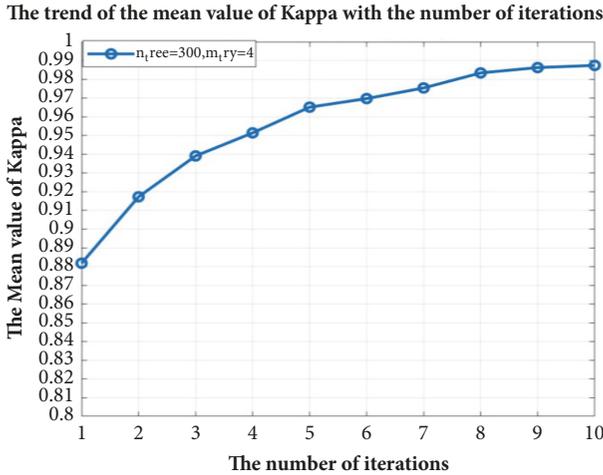


FIGURE 31: Classification kappa trend of different initial and iterative condition of Hyperion data for method 1.

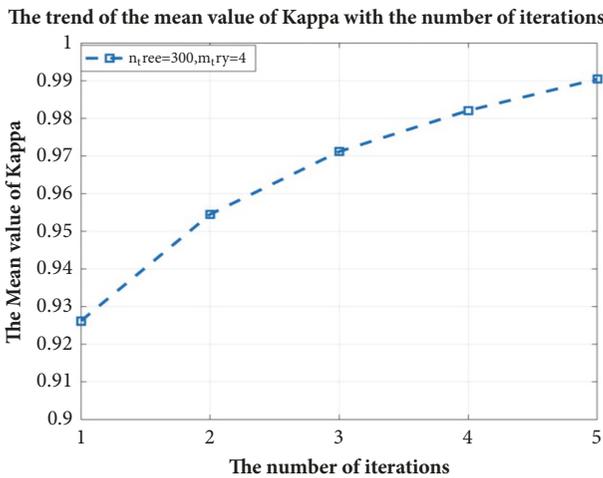


FIGURE 32: Classification kappa trend of different initial and iterative condition of Hyperion data for method 2.

classification accuracy and the kappa coefficient successfully; furthermore, when number of the labeled labels is 5% of the total, the overall accuracy of the AVIRIS data increased to 85.35%, which is about 20% higher than the minimum distance classification algorithm and 3% higher than that of the SVM classification algorithm, what is more, 7.02% higher than that of CART algorithm. The kappa coefficient of the AVIRIS data increased to 0.8591, which is about 0.22 higher than the minimum distance classification algorithm and 0.25 higher than that of the SVM classification algorithm, what is more, 0.08 higher than that of CART algorithm; when number of the labeled labels is 10% of the total, the overall accuracy of the AVIRIS data increased to 87.36%, which is about 22.15% higher than the minimum distance classification algorithm and 5.01% higher than that of the SVM classification algorithm, what is more, 9.03% higher than that of CART algorithm. The kappa coefficient of the AVIRIS data increased to 0.8591, which is about 0.2454 higher

than the minimum distance classification algorithm and 0.055 higher than that of the SVM classification algorithm, what is more, 0.1 higher than that of CART algorithm.

When number of the labeled labels is 5% of the total, the overall accuracy of the Hyperion data increased to 98.83%, which is about 7.88% higher than the minimum distance classification algorithm and 3.92% higher than that of SVM classification algorithm, what is more, 4.71% higher than that of CART algorithm. The kappa coefficient of the Hyperion data increased to 0.9788, which is about 0.1836 higher than the minimum distance classification algorithm and 0.0959 higher than that of SVM classification algorithm, what is more, 0.0887 higher than that of CART algorithm; when number of the labeled labels is 10% of the total, the overall accuracy of the Hyperion data is up to 99.17%, which is about 8.22% higher than the minimum distance classification algorithm and 4.26% higher than that of SVM classification algorithm, what is more, 5.06% higher than that of CART algorithm. The kappa coefficient of the Hyperion data increased to 0.9904, which is about 0.1952 higher than the minimum distance classification algorithm and 0.1075 higher than that of SVM classification algorithm, what is more, 0.1003 higher than that of CART algorithm.

To sum up, the algorithm proposed in this paper can effectively improve the effect of classification. It is very convenient and fast compared with the minimum distance classification algorithm, the SVM, and regression trees (CART) classification algorithm in the same conditions.

4. Conclusions

After experimenting with two different data sources using the proposed method, the following conclusions can be drawn: through a large number of experiments, a set of optimal combination parameters suitable for random forest was obtained. That is, we set the number of decision trees at 300 and the number of nodes at 4. When the random forest parameters were optimal, samples of 5% and 10% of the total were selected for the experiment, and the samples of 5% and 10% were added each time. The weighted entropy algorithm was used to select samples with the largest entropy values to train the new training set proposed in this paper. The classifier used the remaining data as the test data to evaluate the performance of the classifier and to test the universality of the classifier. Compared with the traditional classifier based on supervised classification and SVM, we proved via a large number of experiments that the proposed weighted entropy semisupervised ensemble classifier based on random forest showed better classification performance and better universality.

However, the new algorithm still has inadequacies, such as its long running time and need for more computing power and hardware. Our future research will focus on optimizing the algorithm, lowering its running time, and improving its efficiency.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

TABLE 3: The comparison of overall accuracy and Kappa coefficient under different classifier algorithm of AVIRIS data.

Comparative classification method	Run time (seconds)			Overall Accuracy (%)			Kappa Coefficient		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
Minimum distance classification algorithm	7	10	11	65.46	63.06	65.21	0.6164	0.5895	0.6137
Support vector machine classification algorithm	62	72	86	78.47	80.17	82.35	0.7609	0.7704	0.8041
Classification and regression trees (CART)	17	21	26	75.04	78.0412	78.3285	0.7227	0.7557	0.7591
This-algorithm classifies(5% of the-labeled labels)	16.21	16.63	17.69	82.98	84.19	85.35	0.8108	0.8234	0.8371
This-algorithm classifies(10% labeled labels)	20.65	21.10	22.72	85.85	86.81	87.36	0.8421	0.8528	0.8591

TABLE 4: The comparison of overall accuracy and Kappa coefficient under different classifier algorithm of Hyperion data.

Comparative classification method	Run time (seconds)			Overall Accuracy (%)			Kappa Coefficient		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
Minimum distance classification algorithm	12	15	14	89.26	89.83	90.95	0.7567	0.7693	0.7952
Support-vector machine classification algorithm	25	27	26	94.55	94.63	94.91	0.8744	0.8752	0.8829
Classification and regression trees (CART)	7.5	7.2	6.7	86.0806	93.3795	94.1106	0.6806	0.8468	0.8901
This-algorithm classifies(5% of the labeled labels)	6.04	6.35	6.97	97.22	98.26	98.83	0.9495	0.9684	0.9788
This-algorithm classifies(10% of labeled labels)	7.16	7.69	8.16	98.20	98.90	99.17	0.9711	0.9820	0.9904

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research is supported by the Natural Science Foundation of Henan Province (182300410111), the Key Research Project Fund of Institution of Higher Education in Henan Province (18A420001), Henan Polytechnic University Doctoral Fund (B2016-13), and the Open Program of Collaborative Innovation Center of Geo-Information Technology for Smart Central Plains Henan Province (2016A002).

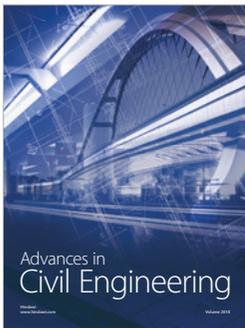
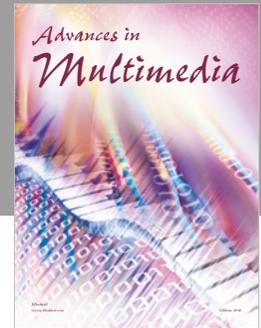
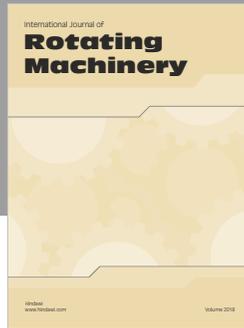
References

- [1] Q. Tong, B. Zhang, L. Zheng et al., *Hyperspectral Remote Sensing-Principle, Technology and Application*, Higher Education Press, Beijing, China, 2006.
- [2] A. F. H. Goetz, "Three decades of hyperspectral remote sensing of the Earth: a personal view," *Remote Sensing of Environment*, vol. 113, supplement 1, pp. S5–S16, 2009.
- [3] Q. Tong, Y. Xue, and L. Zhang, "Progress in hyperspectral remote sensing science and technology in China over the past three decades," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 1, pp. 70–91, 2014.
- [4] G. H. Mitri and I. Z. Gitas, "Mapping post-fire forest regeneration and vegetation recovery using a combination of very high spatial resolution and hyperspectral satellite imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 20, no. 1, pp. 60–66, 2012.
- [5] W. Shuyu, Z. Yuwei, and Y. Zhenhua, "Classification of remote sensing images of honghe wetland based on stochastic forest," *Mapping and Spatial Geography*, pp. 83–85, 2014 (Chinese).
- [6] G. P. Petropoulos, C. Kalaitzidis, and K. Prasad Vadrevu, "Support vector machines and object-based classification for obtaining land-use/cover cartography from Hyperion hyperspectral imagery," *Computers & Geosciences*, vol. 41, pp. 99–107, 2012.
- [7] R. Piiroinen, J. Heiskanen, M. Möttöus, and P. Pellikka, "Classification of crops across heterogeneous agricultural landscape in Kenya using AisaEAGLE imaging spectroscopy data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 39, pp. 1–8, 2015.
- [8] R. R. Nidamanuri and B. Zbell, "Transferring spectral libraries of canopy reflectance for crop classification using hyperspectral remote sensing," *Biosystems Engineering*, vol. 110, no. 3, pp. 231–246, 2011.
- [9] R. Casa, F. Castaldi, S. Pascucci, A. Palombo, and S. Pignatti, "A comparison of sensor resolution and calibration strategies for soil texture estimation from hyperspectral remote sensing," *Geoderma*, vol. 197–198, pp. 17–26, 2013.

- [10] I. Mariotto, P. S. Thenkabail, A. Huete, E. T. Slonecker, and A. Platonov, "Hyperspectral versus multispectral crop-productivity modeling and type discrimination for the HypSPRI mission," *Remote Sensing of Environment*, vol. 139, pp. 291–305, 2013.
- [11] K. Zhao, D. Valle, S. Popescu, X. Zhang, and B. Mallick, "Hyperspectral remote sensing of plant biochemistry using Bayesian model averaging with variable and band selection," *Remote Sensing of Environment*, vol. 132, pp. 102–119, 2013.
- [12] C. Wang, Z. Xu, S. Wang, and H. Zhang, "Semi-supervised classification framework of hyperspectral images based on the fusion evidence entropy," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10615–10633, 2018.
- [13] A. Harris, R. Charnock, and R. M. Lucas, "Hyperspectral remote sensing of peatland floristic gradients," *Remote Sensing of Environment*, vol. 162, pp. 99–111, 2015.
- [14] R. J. Murphy and S. T. Monteiro, "Mapping the distribution of ferric iron minerals on a vertical mine face using derivative analysis of hyperspectral imagery (430–970nm)," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 75, pp. 29–39, 2013.
- [15] K. Tan, R. Zhou, Y. Wan et al., "High-spectral and high-resolution remote sensing monitoring method for underground coal seam combustion," *Journal of Infrared and Millimeter Wave*, vol. 26, pp. 349–358, 2007.
- [16] K. Liu, X. Sun, Z. Zhao et al., "Hyperspectral imaging detection method for ground target camouflage features," *Journal of PLA University of Technology (Natural Science Edition)*, vol. 6, pp. 166–169, 2005.
- [17] W. Lu, X. Yu, Y. Ma, and J. Liu, "Research on detection algorithm of sea warship target by hyperspectral remote sensing image," *Ocean Mapping*, vol. 4, pp. 8–12, 2005.
- [18] Y. Chu, W. Feng et al., *Analysis and Application of Hyperspectral Imagery*, Science Press, Beijing, China, 2013.
- [19] K. Tan, "Research on hyperspectral remote Sensing image classification based on support vector machine," *Mining*, 2010.
- [20] B. Du, L. Zhang, L. Zhang, and W. Hu, "A method for discriminating manifold learning by dimensionality reduction in hyperspectral images," *Photonics newspaper*, vol. 03, pp. 320–325, 2013.
- [21] A. Ghosh, A. Datta, and S. Ghosh, "Self-adaptive differential evolution for feature selection in hyperspectral image data," *Applied Soft Computing*, vol. 13, no. 4, pp. 1969–1977, 2013.
- [22] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2061–2069, 2008.
- [23] B. Bigdeli, F. Samadzadegan, and P. Reinartz, "Fusion of hyperspectral and LIDAR data using decision template-based fuzzy multiple classifier system," *International Journal of Applied Earth Observation and Geoinformation*, vol. 38, pp. 309–320, 2015.
- [24] X. Tang, K. Gao, H. Cheng, and G. Ni, "Hyperspectral unmixing based on ISOMAP and spatial information," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 16, pp. 4283–4287, 2014.
- [25] Z. Bing, "Frontier of hyper-spectral image processing and information extraction," *Journal of remote Sensing*, pp. 1062–1090, 2016.
- [26] F. D. van der Meer, H. M. A. van der Werff, F. J. A. van Ruitenbeek et al., "Multi- and hyper-spectral geologic remote sensing: A review," *International Journal of Applied Earth Observation and Geoinformation*, vol. 14, no. 1, pp. 112–128, 2012.
- [27] Z. Liangpei and L. Jiaye, "Summarization and prospect of sparse information processing of hyper-spectral images," *Acta Sinica remote Sensing*, pp. 1091–1101, 2016.
- [28] P. Dubath, L. Rimoldini, M. Süveges et al., "Random forest automated supervised classification of Hipparcos periodic variable stars," *Monthly Notices of the Royal Astronomical Society*, vol. 414, no. 3, pp. 2602–2617, 2011.
- [29] X. Fan, I. Riaz, Y. Rehman, and H. Shin, "Vanishing point detection using random forest and patch-wise weighted soft voting," *IET Image Processing*, vol. 10, no. 11, pp. 900–907, 2016.
- [30] C. González, J. Mira-McWilliams, and I. Juárez, "Important variable assessment and electricity price forecasting based on regression tree models: Classification and regression trees, Bagging and Random Forests," *IET Generation, Transmission & Distribution*, vol. 9, no. 11, pp. 1120–1128, 2015.
- [31] Y. Liu, P. Du, H. Zheng et al., "Research on classification of domestic small satellite remote sensing image based on random forest science of surveying and mapping," *Science of Surveying and Mapping*, no. 04, pp. 194–196, 2012.
- [32] D.-J. Yu, Y. Li, J. Hu, X. Yang, J.-Y. Yang, and H.-B. Shen, "Disulfide connectivity prediction based on modelled protein 3D structural information and random forest regression," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 3, pp. 611–621, 2015.
- [33] L. Shilei, *Research on stochastic forest algorithm based on Hadoop platform and realization of image classification system*, Xiamen University, 2014.
- [34] M. Zhu, J. Xia, X. Jin et al., "Class weights random forest algorithm for processing class imbalanced medical data," *IEEE Access*, vol. 6, pp. 4641–4652, 2018.
- [35] X. Han-Qiu, "Study on extracting water body information by using improved normalized difference water body index (MNDWI)," *Acta Sinica Sinica*, pp. 589–595, 2005.
- [36] K. Tan, J. Hu, J. Li, and P. Du, "A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 19–29, 2015.
- [37] C. Yi, Z. Xiufang, S. Zhangli et al., "Overview of semi-supervised integrated learning," *Computer Science*, vol. 1, pp. 7–13, 2017.
- [38] M. Belgiu and L. Drăgu, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.
- [39] K. Were, D. T. Bui, Ø. B. Dick, and B. R. Singh, "A comparative assessment of support vector regression, artificial neural networks, and random forests for predicting and mapping soil organic carbon stocks across an Afrotropical landscape," *Ecological Indicators*, vol. 52, pp. 394–403, 2015.
- [40] R.-M. Yang, G.-L. Zhang, F. Liu et al., "Comparison of boosted regression tree and random forest models for mapping topsoil organic carbon concentration in an alpine ecosystem," *Ecological Indicators*, vol. 60, pp. 870–878, 2016.
- [41] I. Nitze, B. Barrett, and F. Cawkwell, "Temporal optimisation of image acquisition for land cover classification with random forest and MODIS time-series," *International Journal of Applied Earth Observation and Geoinformation*, vol. 34, no. 1, pp. 136–146, 2015.
- [42] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS*

- Journal of Photogrammetry and Remote Sensing*, vol. 67, no. 1, pp. 93–104, 2012.
- [43] L. Li, C. Solana, F. Canters, and M. Kervyn, “Testing random forest classification for identifying lava flows and mapping age groups on a single Landsat 8 image,” *Journal of Volcanology and Geothermal Research*, vol. 345, pp. 109–124, 2017.
- [44] J. C.-W. Chan and D. Paelinckx, “Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery,” *Remote Sensing of Environment*, vol. 112, no. 6, pp. 2999–3011, 2008.
- [45] V. F. Rodriguez-Galiano, M. Chica-Olmo, F. Abarca-Hernandez, P. M. Atkinson, and C. Jeganathan, “Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture,” *Remote Sensing of Environment*, vol. 121, pp. 93–107, 2012.
- [46] C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu, “Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas,” *Remote Sensing of Environment*, vol. 187, pp. 156–168, 2016.
- [47] L. Zg, *Some studies on stochastic forest improvement*, Xiamen University, 2013.
- [48] M. Śmieja, “Weighted approach to general entropy function,” *IMA Journal of Mathematical Control and Information*, vol. 32, no. 2, pp. 329–341, 2015.
- [49] J. Wang, W. Pang, L. Wang et al., “Synthetic evaluation of steady-state power quality based on combination weighting and principal component projection method,” *CSEE Journal of Power and Energy Systems*, vol. 3, no. 2, pp. 160–166, 2017.
- [50] X. Qu, H. Chen, and G. Peng, “Novel detection method for infrared small targets using weighted information entropy,” *Journal of Systems Engineering and Electronics*, vol. 23, no. 6, pp. 838–842, 2012.
- [51] S. Ghohinejad, R. Shad, H. S. Yazdi, and M. Ghaemi, “Improving Signal Subspace Identification Using Weighted Graph Structure of Data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 6, pp. 831–835, 2017.
- [52] Y. Xu, Y. Wang, and X. Miu, “Multi-attribute decision making method for air target threat evaluation based on intuitionistic fuzzy sets,” *Journal of Systems Engineering and Electronics*, vol. 23, no. 6, pp. 891–897, 2012.
- [53] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, “Robust and Accurate Shape Model Matching Using Random Forest Regression-Voting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1862–1874, 2015.
- [54] N. Segev, M. Harel, S. Mannor, K. Crammer, and R. El-Yaniv, “Learn on Source, Refine on Target: A Model Transfer Learning Framework with Random Forests,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1811–1824, 2017.
- [55] J. Chen, K. Li, Z. Tang et al., “A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 919–933, 2017.
- [56] F. Miao, Y.-P. Cai, Y.-X. Zhang, X.-M. Fan, and Y. Li, “Predictive modeling of hospital mortality for patients with heart failure by using an improved random survival forest,” *IEEE Access*, vol. 6, pp. 7244–7253, 2018.
- [57] X. Wang, “Ladle furnace temperature prediction model based on large-scale data with random forest,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 770–774, 2017.
- [58] C. Wang, Z. Guo, S. Wang, L. Wang, and C. Ma, “Improving hyperspectral image classification method for fine land use assessment application using semisupervised machine learning,” *Journal of Spectroscopy*, vol. 2015, Article ID 969185, 8 pages, 2015.
- [59] W. Chunyang, G. Zengzhang, W. Shuangting et al., “A method of strip noise removal from hyper-spectral images by fusion of bilateral filtering and moment matching,” *Journal of surveying and Mapping Science and Technology*, pp. 153–156, 2014.
- [60] L. Dong, K. Jie, H. Gensheng et al., “Treatment of thick cloud and cloud shadow in remote sensing image based on support vector machine,” *Acta Geodaetica Et Cartographica Sinica*, vol. 41, no. 2, pp. 225–616, 2012.
- [61] N. Payet and S. Todorovic, “Hough forest random field for object recognition and segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1066–1079, 2013.
- [62] X. Wang and C. Chen, “Ship Detection for Complex Background SAR Images Based on a Multiscale Variance Weighted Image Entropy Method,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 2, pp. 184–187, 2017.
- [63] J. Xia, W. Liao, J. Chanussot, P. Du, G. Song, and W. Philips, “Improving Random Forest With Ensemble of Features and Semisupervised Feature Extraction,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 7, pp. 1471–1475, 2015.
- [64] P. Kotschieder, S. R. Bulò, M. Pelillo, and H. Bischof, “Structured Labels in Random Forests for Semantic Labelling and Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2104–2116, 2014.
- [65] J. L. Alba, L. Docio, and S. Ruibal, “Soft-competitive-growing classifier with unsupervised fine-tuning,” in *Proceedings of the 1997 IEEE International Conference on Neural Networks, ICNN 1997*, vol. 3, pp. 1418–1423, June 1997.
- [66] N. K. Anh, N. Van Linh, N. K. Toi, and N. T. Tarn, “Multi-labeled document classification using semi-supervised mixture model of Watson distributions on document manifold,” in *Proceedings of the 2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*, pp. 123–128, December 2013.
- [67] O. Rajadell, P. García-Sevilla, V. C. Dinh, and R. P. W. Duin, “Improving hyperspectral pixel classification with unsupervised training data selection,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 3, pp. 656–660, 2014.
- [68] F. Lin and W. W. Cohen, “Semi-supervised classification of network data using very few labels,” in *Proceedings of the 2010 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2010*, pp. 192–199, August 2010.
- [69] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, “Incremental Learning of Random Forests for Large-Scale Image Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 490–503, 2016.
- [70] H. Pang, S. L. George, K. Hui, and T. Tong, “Gene selection using iterative feature elimination random forests for survival outcomes,” *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 5, pp. 1422–1431, 2012.
- [71] N. Quadrianto and Z. Ghahramani, “A very simple safe-Bayesian random forest,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1297–1303, 2015.
- [72] V. E. Kosmidou, P. C. Petrantonakis, and L. J. Hadjileontiadis, “Enhanced Sign Language Recognition Using Weighted Intrinsic-Mode Entropy and Signer’s Level of Deafness,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1531–1543, 2011.

- [73] X. Ma, J. Guo, K. Xiao, and X. Sun, "PRBP: Prediction of RNA-Binding Proteins Using a Random Forest Algorithm Combined with an RNA-Binding Residue Predictor," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 6, pp. 1385–1393, 2015.
- [74] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.
- [75] C. C. M. Chen, H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and P. MacRossan, "Methods for identifying SNP interactions: a review on variations of logic regression, random forest and Bayesian logistic regression," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 6, pp. 1580–1591, 2011.
- [76] A. Topirceanu and G. Grosseck, "Decision tree learning used for the classification of student archetypes in online courses," *Procedia Computer Science*, vol. 112, pp. 51–60, 2017.
- [77] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.
- [78] J. Wang, M. Li, Y.-T. Hu, and Y. Zhu, "Comparison of hospital charge prediction models for gastric cancer patients: neural network vs. decision tree models," *BMC Health Services Research*, vol. 9, no. 1, 2009.
- [79] I. Chikalov, S. Hussain, and M. Moshkov, "Bi-criteria optimization of decision trees with applications to data analysis," *European Journal of Operational Research*, vol. 266, no. 2, pp. 689–701, 2018.



Hindawi

Submit your manuscripts at
www.hindawi.com

