IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2019, accepted July 28, 2019, date of publication August 2, 2019, date of current version August 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932909

Research on Optimization Methods of ELM Classification Algorithm for Hyperspectral Remote Sensing Images

FANG HUANG[1], JUN LU[1], JIAN TAO[2], LI LI[1], XICHENG TAN[3], AND PENG LIU[4]

[1]School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Texas Engineering Experiment Station, High Performance Research Computing, and Texas A&M Institute of Data Science, Texas A&M University, College Station, TX 77843, USA
[3]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[4]Aerospace Information Research Institute (AIR), Chinese Academy of Sciences (CAS), Beijing 100094, China

Corresponding authors: Fang Huang (fang.percy.huang@gmail.com) and Peng Liu (liupeng202303@aircas.ac.cn)

This work was supported mainly in part by the National Key Research and Development Program of China under Grant 2018YFC1505205, in part by the Hubei Provincial Key Laboratory of Intelligent Geo-Information Processing (China University of Geosciences) under Grant KLIGIP-2017A06, Grant KLIGIP-2017A07, and Grant KLIGIP-2017A09, in part by the Shanghai Aerospace Science and Technology Innovation Fund under Grant SAST2016006, and in part by the Beijing Key Laboratory of Urban Spatial Information Engineering under Grant 2017209.

ABSTRACT In land-use classification of hyperspectral remote sensing (RS) images, traditional classification methods often experience large amount of datasets and low efficiency. To solve these problems, a fast machine-learning method, the extreme learning machine (ELM) algorithm, was introduced. However, basic use of the ELM usually encounters problems of unstable classification results and low classification accuracy. Hence, in this paper, optimization methods for ELM-based RS image classification were mainly discussed and applied to solve the bottleneck problems. From the three perspectives of ensemble learning, making full use of image texture features, and deep learning, three classification optimization methods have been designed and implemented. The results show that: 1) To some extent, all the three methods can achieve a balance between classification accuracy and efficiency, i.e., they can maintain the advantage of ELM algorithm in classification efficiency and speed while have better classification accuracy; 2) The image texture feature optimization method (LBP-KELM) solves the problem of unsatisfactory classification results experienced by the ensemble learning optimization method (Ensemble-ELM) and further improves classification accuracy. However, the classification results are sensitive to the type of dataset; and 3) Fortunately, the optimization method combined with deep learning (CNN-ELM) can meet the application needs of multiple datasets. Furthermore, it can also further improve classification accuracy.

INDEX TERMS Hyperspectral remote sensing, ELM algorithm, ensemble learning, texture features, deep learning.

I. INTRODUCTION

The use of remote sensing (RS) images for land-use coverage classification is an important part of obtaining land-cover information and is also a key topic in current land-use/land-cover change research. In recent years, due to their advantages such as multiple bands, high-resolution, and rich information, hyperspectral RS images have been incorporated into land-use classification and have become a research hotspot in the RS field [1]–[7]. However, the massive and high-dimensional features of hyperspectral RS data have also posed great challenges to RS image classification research. When traditional classification methods, e.g., iterative self-organizing data analysis techniques (ISODATA), artificial neural networks (ANN), and support vector machines (SVM), are used in hyperspectral RS classification applications, they are likely to involve excessive processing data scale, high computational complexity, and vulnerability to local minima. In particular, these methods also have lower classification efficiency and speed, which means that they have difficulty in meeting the needs of current applications [8]–[14].

Consequently, some researchers are studying hyperspectral RS classification problems using novel theory and method,

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

e.g., multivariate statistics or deep learning methods etc. Especially, Borhani and Ghassemian proposed a new concept of spectral-spatial kernel-based multivariate analysis based on the statistical principle of multivariate statistics, KMVSSA, for dimensionality reduction, feature selection and classification on hyperspectral remotely sensed datasets. They states that the proposed KMVSSA framework is to expose the inherent structure and meaning revealed within spectral and spatial features through various statistical methods. Through the systemically experiments with other state of the art classification methods, it show that the KMVSSA framework can greatly improves the classification accuracies with hyperspectral RS images [15]. Nowadays, more and more researchers are studying these problems using deep learning methods. For example, Wang *et al*. introduced a hybrid of principle component analysis (PCA), guided filtering, and deep learning architecture for hyperspectral RS data classification. In this approach, both spatial and spectral features are efficiently explored to construct feature vectors. The proposed method promotes hyperspectral data classification accuracy by introducing filtering of local pixel information and using multiple features for deep feature learning [16]. In another study, Liu *et al*. investigated sparse auto-encoders (SAE) and SVM to examine their sensitivity. Then they conducted a systematic comparative evaluation [17]. The extreme learning machine (ELM) is a rapid machine-learning algorithm proposed by Huang (2006). It exhibits high learning speed, high efficiency, and good generalization performance when dealing with large-scale data [18], [19]. Therefore, it can be foreseen that using the ELM algorithm for hyperspectral RS image classification will effectively overcome the bottlenecks facing the field.

From literatures review, it found that in recent years, the ELM algorithm has begun to be applied to land-use classification with hyperspectral RS images. This method has achieved good results in this context because it has many advantages such as fast processing speed and good generalization performance. However, relevant studies are still in the initial stage, and more efforts are needed to exploit ELM's potential fully. Based on the respective advantages of ELM algorithms and hyperspectral RS images, it is suitable to propose further performance optimization work for ELM-based classification of hyperspectral RS images. Such improvement and optimization methods have much room for development in this area. To realize this potential, in this study, land-use optimization methods for hyperspectral RS imagery based on the ELM algorithm have been fully developed and used based on three aspects: (1) The leading-edge theory of ensemble learning; (2) Exploring the rich spatial texture features of hyperspectral imagery; and (3) Deep learning technology.

(1) In order to solve the problem of instability of classification results in ELM algorithm in RS image classification, we design and implement a new ELM RS image classification method based on ensemble learning, and carry out the ELM RS image classification process (i.e., Ensemble-ELM). Improvement and optimization were achieved by modifying the ELM-based classifier to enhance the robustness of the algorithm. We combined the voting method and maximum probability method, so that the classification results are more accurate. The experimental results show that the method effectively solves the instability of ELM classification, and the classification results are also improved. However, there are still rooms for further improvement as there are still many pitting points in the classification map.

(2) To overcome the defects mentioned in (1) above, and to further improve the classification results and the classification accuracy, this research considers and utilizes the rich spatial texture features of the images, and designs the local binary pattern (LBP) based kernel ELM (KELM), i.e., LBP-KELM RS image classification method [20]. Firstly, the minimum noise fraction (MNF) is used to reduce the dimension of the band. Then the LBP operator is used to extract the rich texture features of the RS image, and finally the radial basis function is used as the kernel function to construct the KELM classifier. In the specific implementation process of the LBP-KELM method, it is very important to study the influence of the setting of each parameter on the classification result. The optimal value of each parameter is determined by several experiments for classification, and good results are obtained. The disadvantage of this method is that the classification results are more sensitive to the data set type.

(3) In order to improve the classification accuracy and meet the classification requirements of various data sets (existed problem in LBP-KELM), this research takes advantage of deep learning to optimize the ELM RS image classification, and designs a classification model that combines convolutional neural networks (CNN) and ELM (i.e., CNN-ELM). The convolution and subsampling layers are alternately connected to construct the depth feature extraction layer, and the ELM is used to construct the classification layer. Our implementation is based on the current popular deep learning framework Keras.[1] The experimental results demonstrate the advantages of the method in classification accuracy.

In summary, this research will be helpful to provide future researchers with comprehensive analyses of land-use classification using hyperspectral RS images.

The paper is organized as follows. Section 2 discusses the related works of the improvement and optimization measures used with ELM for hyperspectral RS image classification. Section 3 gives a brief introduction to the principles and process of the ELM algorithm-based classification methodology. Section 4 concentrates mainly on the design and implementation of the proposed ELM-based land-use cover classification optimization methods for hyperspectral RS images from the aspects of ensemble learning, making full use of image texture features, and deep learning. Section 5 presents various experiments performed to verify the accuracy and performance of the three optimization methods on different hyperspectral RS datasets and discusses the experimental results.

[1] https://keras.io/

Finally, Section 6 draws some conclusions and points out future research directions.

II. RELATED WORKS OF ELM ALGORITHM

In recent years, ELM has been used increasingly for hyperspectral RS image classification. This method can effectively solve the problems of complex computations and excessive processing data scale caused by the multi-spectral nature of hyperspectral RS data, thus making RS image classification more efficient. Pal used the basic ELM directly for land-cover classification of hyperspectral RS images and compared the classification results obtained with those from a backpropagation (BP) neural network. The results show that ELM can achieve the same classification accuracy as a BP neural network, but with much higher classification speed [21]. However, it still encounters problems such as algorithm instability, lack of robustness, and low classification accuracy [22]. As a result, it is not satisfactory to use the basic ELM algorithm directly for RS image classification.

Therefore, researchers have taken various perspectives and combined various leading-edge technologies to improve and optimize the ELM classification method. These measures intend to improve classification accuracy while ensuring ELM's advantages in classification efficiency, thus making the ELM classification method more extensively applicable.

On the one hand, to improve the instability of the classification results caused by the input weights of the ELM algorithm and its random hidden-layer bias assignments, researchers mostly use ensemble learning ideas to optimize the ELM-based RS image classification method to enhance the robustness of the algorithm. The reason for this is that ensemble learning offers characteristics like generalization and stability, which should be useful for integration with unstable ELM classifiers. For example, Samat *et al*. proposed two ELM RS image classification methods based on ensemble learning in 2014 [22]. They integrated the ELM-based classifiers using commonly used ensemble learning methods, Bagging and AdaBoost. A comparison of this method using ELM and SVM with experiments on multiple hyperspectral RS images shows that applying ensemble learning to ELM classification can compensate for the deficiencies of random input weight generation in the ELM algorithm and enhance algorithm robustness. Therefore, this method achieves better classification accuracy (over 3% better) than ELM, which is equivalent to the SVM method. Although the improvement is not obvious, the classification speed of the combined approach still maintains the ELM algorithm's significant advantages in classification efficiency, which are far beyond those of the SVM method. Han *et al*. also used the ensemble learning idea to optimize the ELM classification method [23]. They integrated multiple ELM classifiers and used a simple voting method to vote on the prediction category of each classifier to obtain the final classification result. With experiments on Zhalong Wetland and Haicheng City RS images, they showed that the overall classification accuracy for the two regions was improved by 5.1% and 9.3%, respectively, compared to the ELM method. Clearly, the overall classification accuracy gain from adopting the ensemble learning optimization method can improve classification accuracy and effectiveness [23]. The preceding discussion has shown that existing studies of optimizing ELM classification using ensemble learning simply combines commonly used ensemble learning algorithms directly with ELM. As a result, classification effectiveness is not much improved. In addition, to alleviate the instability of ELM classification results, some researchers have also proposed new image classification methods based on KELM, i.e., replacing the hidden layer of ELM with a kernel function so that the input weights do not need to be randomly generated. These methods not only enhance the stability of the classification results, but also have lower computational complexity [24]–[27].

On the other hand, RS images contain abundant spatial information such as spatial neighborhood features and texture features. To make full use of this abundant spatial information to improve ELM classification results, Lv *et al*. proposed an RS classification method based on hierarchical local-receptive-field ELM (HL-ELM), which fully considered the local spatial characteristics of RS data and used a multi-layer ELM structure. By comparing the experiments using SVM, basic ELM, and other methods, the classification results of the proposed method were found to be optimal while still maintaining the classification speed advantage of the ELM algorithm [28]. Cao *et al.* used a circular belief propagation algorithm to calculate the spatial correlation between pixels for classification and thus improved ELM classification performance with RS images [29]. Chen *et al*. used Gabor filters to capture certain physical structures, such as directional information, in RS images to extract useful spatial features for classification. Experiments showed that classification accuracy was greatly improved and that the classification results were very satisfactory [24]. Li *et al*. extracted texture features such as edges and corners of RS images for ELM classification and successfully obtained better classification results than traditional methods [20].

In recent years, some researchers have tried to combine deep learning with ELM for RS image classification. In their studies, they used deep learning algorithms to extract depth features from images to obtain more accurate classification results. For instance, Lv *et al*. proposed an in-depth ELM model. The improved model was divided into two main phases: (1) An ELM-based automatic encoder for feature extraction, and (2) An ELM classifier. Through comparative experiments with basic ELM, SVM, and other classification methods, this method can accurately extract the spatial distribution of RS images and greatly improve classification accuracy [30]. Lv *et al*. also used deep learning to extract the depth features of RS images and combined it with ensemble learning to optimize the ELM classification algorithm, thus achieving better classification results [31]. However, at present, sufficient relevant research is still lacking to optimize ELM classification of RS images with deep learning. With the rapid development of deep learning theory,

major breakthroughs are expected in this area in the next few years.

III. PRINCIPLE OF THE ELM ALGORITHM AND THE ELM-BASED CLASSIFICATION PROCESS

A. BASIC PRINCIPLE OF THE ELM ALGORITHM

The ELM algorithm was built upon the basic model of the single-hidden-layer feedforward neural network (SLFN). The following description covers the network structure and working principles of the SLFN.

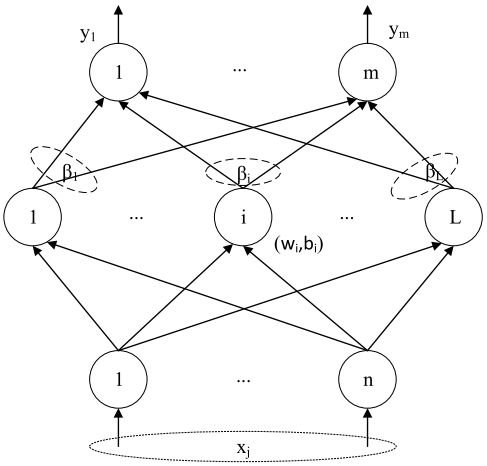


FIGURE 1. Network structure of an SLFN.

Suppose that there is a dataset (x_i, y_i), i $= 1, 2 \ldots, N$, where $x_i = [x_{i1}, x_{i2}, \ldots x_{in}]^T \in R^n$, that represents the sample characteristics; $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T \in R^m$ represents the labels of sample i in m categories, and $y_{ij} \in \{0, 1\}, j = 1, 2, \ldots$ m. Figure 1 shows the network structure of an SLFN with L hidden nodes ($L \leq N$) and the activation function $g(x)$. The corresponding mathematical model of an SLFN is expressed in Eq. (1).

$$y_j = \sum_{i=1}^{L} \beta_i g(w_i, b_i, x_i), \quad j = 1, 2, \ldots N \tag{1}$$

where β_i denotes the output weight of the i-th hidden layer node to the output layer; w_i and b_i are randomly generated, w_i represents the input weight of the i-th node of the hidden layer, b_i represents the biases of the i-th hidden layer node; $g(w_i, b_i, x_i)$ represents the activation function of the i-th node in the hidden layer and the way to connect to the output layer. Equation (1) can be simplified as follows:

$$H\beta = Y \tag{2}$$

where

$$H = H(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_N) = \begin{bmatrix} g(w_1, b_1, x_1) & \cdots & g(w_L, b_L, x_1) \\ \vdots & \cdots & \vdots \\ g(w_1, b_1, x_N) & \cdots & g(w_L, b_L, x_N) \end{bmatrix}_{N \times L}$$

represents the hidden layer output matrix [32] and

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}$$

are as defined above.

The parameters of the SLFN can be calculated using the least-squares solution:

$$min\|H\beta - Y\| \tag{3}$$

In other words, ELM needs to find a set of optimal parameters $\hat{\beta}_i, \hat{w}_i, \hat{b}_i, i = 1, 2 \ldots, L$ so that the following equation holds:

$$\|H(\hat{w}_1, \ldots, \hat{w}_L, \hat{b}_1, \ldots, \hat{b}_L)\hat{\beta} - Y\| = \min_{\beta, w_i, b_i} \|H(\hat{w}_1, \ldots, \hat{w}_L, \hat{b}_1, \ldots, \hat{b}_L)\beta - Y\| \tag{4}$$

The least-squares solution of this equation is:

$$\hat{\beta} = H^+ Y \tag{5}$$

where H^+ is the M-P generalized inverse of the matrix H [33].

B. ELM-BASED CLASSIFICATION ALGORITHM PROCESS AND STEPS

From the above discussion, the input weights w_i and the hidden layer biases b_i are set randomly at the beginning of the training process, and the output layer weights β_i can be obtained by the above formula $\hat{\beta} = H^+ Y$. Suppose that the known training samples have a number of hidden layer nodes L and an activation function $g(x)$. Then the learning process of the ELM neural network can be roughly described as in Table 1.

TABLE 1. Process of ELM neural network learning algorithm.

Algorithm: ELM
Input：sample set $\{(x_i, y_i) \mid x_i \in R^n, y_i \in R^m, i = 1, 2, ..., N\}$, the number of hidden layer nodes L, activation function $g(x)$
Output：output weight β
1) Randomly generate the values for input weight w_i and hidden layer offset b_i;
2) Calculate hidden layer output matrix H;
3) Calculate output weight β: $\hat{\beta} = H^+ Y$.

According to Table 1, the basic ELM-based RS image classification process can be divided into three steps: (1) the spectral features of RS images are extracted to construct a training sample set; (2) the ELM classifier is trained; and (3) the unknown sample is classified and tested to output the final prediction label of the sample. Figure 2 shows the specific process.

In this process, the training and testing of the ELM classifier are the two most important steps. The training of ELM obtains the output weights β. The testing of ELM classifies

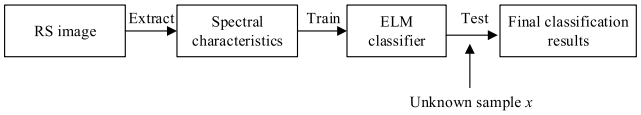


FIGURE 2. Processing flowchart of the ELM-based classification algorithm.

the test set and obtains its prediction labels using the parameters determined during the training phase.

IV. OPTIMIZATION APPROACHES USING THE ELM ALGORITHM FOR HYPERSPECTRAL RS IMAGE CLASSIFICATION

To explore fully the high stability, high precision, and data type adaptability of the ELM classification algorithm, in this study, three classification optimization methods were designed and implemented from the perspectives of ensemble learning, making full use of image texture features, and deep learning respectively.

A. OPTIMIZATION APPROACH 1: INTEGRATING ELM WITH ENSEMBLE LEARNING (ENSEMBLE-ELM)

The core concept of ensemble learning is to train multiple base classifiers and to use a certain combination strategy to combine the output results of each base classifier to obtain the final classification category of the sample, an approach that can provide better performance than a single base classifier. When ELM is applied directly to hyperspectral RS image classification, it encounters problems such as instability and lack of robustness. To solve such problems and improve algorithm robustness, certain strategies must be adopted to optimize the algorithm. Fortunately, ensemble learning has the characteristics of strong generalization performance and high stability. It is especially effective for unstable integrated classification with weak classifiers, meaning that it is perfectly suited for the unstable ELM algorithm. For this reason, this study integrated ensemble learning into the ELM algorithm (for the sake of simplicity, the combination was called Ensemble-ELM). Theoretically, the integrated ELM algorithm with multiple ELM-based classifiers would improve generalization performance and classification accuracy.

1) OVERALL CLASSIFICATION STRATEGY AND IMPLEMENTATION STEPS OF ENSEMBLE-ELM

Figure 3 shows the overall classification framework of Ensemble-ELM for hyperspectral RS image classification.

Figure 3 reveals that the Ensemble-ELM classification method consists of five steps.

(1) First, the spectral characteristics of each pixel are extracted from the hyperspectral RS image. Because the number of image bands is large, dimension reduction or band selection should generally be carried out first to reduce the influence of the noise band on the classification result.

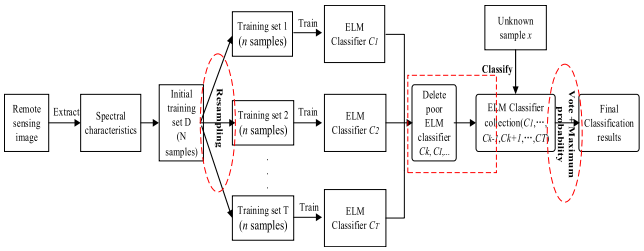


FIGURE 3. Overall classification framework of the Ensemble-ELM method for hyperspectral RS image classification.

Then N typical samples are selected to constitute the initial training set D.

(2) Resampling methods are used to construct T training sets. To do this, n ($n < N$) samples are randomly extracted from D and then put back. Thus, one training set is formed. The total number of cycles in this operation is T.

(3) The base classifier is constructed using the ELM algorithm. Then the ELM classifiers are trained with T training sets, where the size of each training set is n, and T prediction results are obtained.

(4) Based on the classification prediction results of the T ELM-based classifiers, a certain strategy is used to delete ELM-based classifiers with poor or unstable classification results. Assuming that the number of base classifiers to be deleted is M ($T > M \geq 1$), the remaining $(T - M)$ classifiers will be combined to be used as the base classifier set.

(5) For the sample dataset with unknown class labels, the samples in the dataset are classified using the $(T - M)$ residual base classifiers in step (4). Thus, each sample is the object of $(T - M)$ classification results. However, the final class label of each unknown class dataset will be determined by a combination of the voting method and the maximum probability method.

In summary, the Ensemble-ELM algorithm proposed in this study uses the ensemble learning concept to improve the ELM classification method from the point of view of overall processing, an approach that is similar to existing studies. In particular, three detailed aspects of this overall framework were chosen for optimization; they are marked with a red dotted frame in Fig. 3. The following discussion will explore the design and implementation of these specific optimization points.

2) RESAMPLING THE TRAINING DATASET

The main step in training set resampling is to extract and put out n ($n < N$) samples randomly from the initial training set D containing N samples to form a training set for T times and thus construct T different training sets. The number of iterations of T must be manually set. In this study, the value of T circle was set to a better value through multiple experiments. Such a set can achieve a balance between classification accuracy and efficiency, which can better meet the application's accuracy and efficiency requirements. Figure 4 shows a process flowchart for generating T training sets.

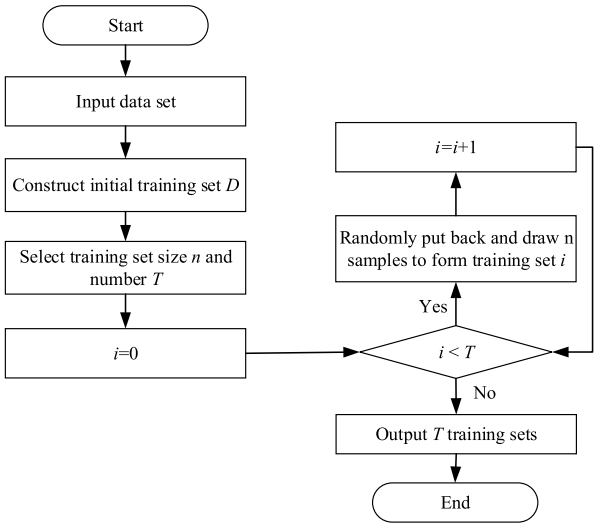


FIGURE 4. Flow diagram of generating T training sets with the resampling method.

3) DESIGN AND IMPLEMENTATION OF PRUNING THE ELM-BASED CLASSIFIER METHOD

To optimize the performance of ELM integration classification, a method of pruning ELM-based classifiers was designed. This method can be divided into two main steps: (1) Unstable ELM-based classifiers are deleted, and (2) The poorest of the remaining ELM-based classifiers is removed.

(1) Deleting unstable ELM-based classifiers. As mentioned above, the ELM algorithm is an unstable algorithm, i.e., the accuracy of the classification result is extremely low, occasionally less than 10% or 50%. Such unstable classification results will lead to large errors in training sample classification accuracy. If unstable classification results occur in T ELM-based classifiers during training, they will compromise performance and accuracy during integration of the T base classifiers. Therefore, while training the ELM-based classifiers 10 times using T training sets, any ELM-based classifiers with extremely low classification accuracy should be deleted, and the remaining ELM-based classifiers should be combined to classify new datasets. A threshold value a must be set for deleting an unstable ELM-based classifier. When the classification accuracy is less than a, the ELM-based classifier is deleted. The value of the threshold a determined experimentally in this study was 50%.

(2) Deleting the worst ELM classifier. After deleting unstable ELM classifiers, the next step is to find and delete the worst ELM classifier. The theoretical basis of this operation depends on ensemble learning theory: it assumes that the important factor in improving ensemble learning performance is the difference between the base classifiers. When a base classifier is deleted, the error obtained by integrating the base classifiers is smaller than that obtained by deleting other base classifiers. In this case, it can be concluded that the base classifier can improve the difference in residual-base learner integration compared with other base classifiers. Therefore, the method to evaluate the merits of an ELM-based classifier is to delete one ELM-based classifier first, then integrate the remaining ELM-based classifiers, and calculate the error of the integrated classifiers. This error is used to compare the advantages and disadvantages of different sets of integrated ELM-based classifiers and to determine which ELM-based classifier among them is the worst and should be deleted. These steps should be repeated, deleting a different classifier each time, until all base classifiers have been experimentally deleted. The ELM-based classifier set with the smallest error can then be identified.

TABLE 2. Algorithm implementation for pruning the ELM-based classifier.

Algorithm: Pruning ELM-based classifiers
1) Input the T training sets resampled from the initial training set;
2) for i=1: T
3) for j=1:10
4) train ELM base classifier;
5) if classification accuracy < threshold a
6) break;
7) end
8) delete this ELM base classifier
9) end
10) the remaining classifiers form the base classifier set $CSEN_0$;
11) for i=1:N-1
12) for j=1 : (N+1-i)
13) delete the j-th ELM base classifier C_{ij} from $CSEN_{i-1}$ (C_{ij} represents the j-th ELM-base classifier in the i-th round);
14) combine the remaining ELM-based classifiers for integration and then calculate errors e_{ij};
15) put this ELM-based classifier back;
16) end
17) $e_{im} = \min\{e_{ij}, j = 1,...,(T+1-i);\}$; record the value of m;
18) delete the m-th ELM base classifier, the remaining ELM-based classifiers are combined into sets $CSEN_i$;
19) the error of this round is recorded as E_i;
20) end
21) error output $E_y = \min\{E_i, i = 1,...,(T-1);\}$; record the value of y;
22) the base classifier's output is $CSEN_y$

Table 2 describes the specific implementation procedure for pruning the ELM-based classifier.

4) OPTIMIZING THE COMBINATION OF ELM-BASED CLASSIFIERS

In ensemble learning, a simple voting method is generally used to integrate the base classifiers. In this voting method, the final classification results for the samples are determined by the voting of each base classifier. However, because this simple method determines the classification result through a ''one-person-one-vote'' rule, without considering the differences between the various base classifiers, the integration results are unremarkable. Therefore, to improve the accuracy of ELM integration classifiers, this paper combines the simple voting method with the maximum probability method to integrate the ELM classifiers, so that the two methods provide complementary benefits. Figure 5 describes specific steps of this procedure.

According to Fig. 5, the main steps of this method are:

(1) First, the voting results of the T ELM-based classifiers are evaluated using the simple voting method. If the voting

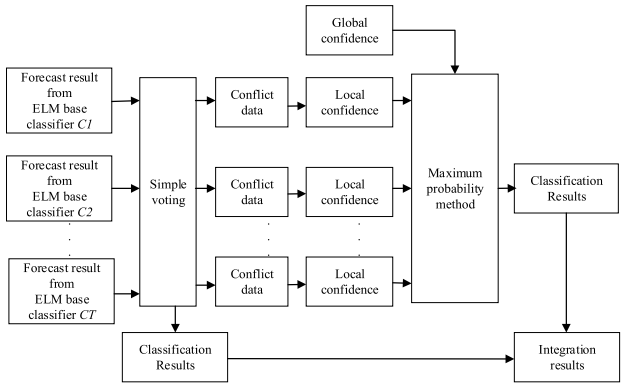


FIGURE 5. Flowchart of ELM-based classifier integration using a simple voting method and a maximum probability method.

result is 100%, the classification results of all the base classifiers are considered to be the same, and the samples are directly classified into the chosen category.

(2) If the voting results are inconsistent, then the final classification result is determined according to local confidence. If only one local confidence is higher than the given threshold c, it will be directly used as the final prediction result. If two or more local confidences are higher than the threshold c, or if all the local confidences are lower than c, the global confidence level is used to determine the result. This involves selecting the base classifier with the highest classification accuracy among all the base classifiers as providing the final prediction category of the sample. In this study, the value of the threshold c was determined to be 85% after several experiments.

(3) Finally, the final integrated classification results of all the samples are judged by combining the classification result of the simple voting method with the classification result of the maximum probability method.

5) ANALYSIS OF THE ENSEMBLE-ELM OPTIMIZATION METHOD

The Ensemble-ELM optimization method proposed in this paper can not only provide the RS image classification efficiency of the ELM algorithm, but can also enhance the algorithm's robustness and improve its classification accuracy. However, there are still many pits in the classification result map, and the results are still not satisfactory with specific applications. The main reason for the unsatisfactory classification results is that the traditional ELM classification methods and the optimized method with ensemble learning, i.e., Ensemble-ELM, both use spectral features to classify RS images. When working with spectral features, the same object can appear to have different spectral characteristics, and different objects can appear to have the same spectral characteristics. This can result in misclassifications and missing points and large quantities of "salt and pepper noise". Another reason for the unsatisfactory classification result is the low classification accuracy due to not making full use of the rich spatial information in hyperspectral images. Texture features are an important spatial information component of RS images. They can effectively resolve cases that appear to be the same object with different spectral characteristics or different objects with the same spectral characteristics and fully account for the macroscopic and microscopic structure of RS images. Many studies have shown that using texture feature-based classification methods can yield better classification results. Therefore, to improve ELM classification performance, this study fully considered the impact of rich spatial texture features on classification and proposed a KELM RS image classification optimization method based on LBP texture features, which will be fully discussed in the following section.

B. OPTIMIZATION APPROACH 2: INTEGRATING KERNEL ELM WITH LBP TEXTURE FEATURES (LBP-KELM)

This section investigates the ELM algorithm with the LBP texture feature method to optimize the basic ELM algorithm. To improve the performance of the ELM algorithm, the radial basis function (RBF) kernel function was used instead of the activation function in the basic ELM algorithm to construct the KELM classifier. Through this replacement, the improved ELM algorithm does not need to randomly generate the input weights, which can not only enhance the stability of the classification results, but also reduce the computational complexity of the algorithm. As a result, the single hidden layer feedforward neural network does not need to update the input weights. Here, this optimization method can be abbreviated as the LBP-KELM method. Figure 6 shows a flowchart of the LBP-KELM classification method.

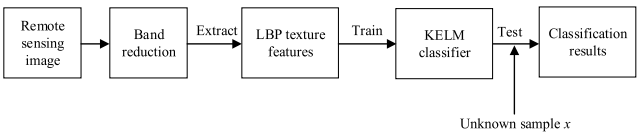


FIGURE 6. Flowchart of the LBP-KELM hyperspectral RS image classification algorithm.

Figure 6 shows that using the LBP-KELM optimization method in hyperspectral RS image classification involves three main steps: (1) Reduce band dimensionality; (2) Extract spatial texture features with the LBP algorithm; and (3) Construct the KELM classifier. This study has carried out optimization corresponding to these three aspects.

1) IMPLEMENTATION OF MNF BAND DIMENSION REDUCTION METHOD

The MNF method can solve the problems of excessively large computation requirements and low accuracy of classification results due to redundant information interference caused by multi-spectral images of hyperspectral RS images. The MNF can also suppress detrimental interference with image quality caused by noise. In fact, MNF is an improvement to the principal component analysis (PCA) method. It uses the signal-to-noise ratio (SNR) as an indicator to describe image quality

and effectively separates noise from actual data. MNF is, in principle, a principal component analysis transformation that involves two overlapping processes. The specific steps are as follows:

(1) Estimate the noise covariance matrix C_N of the RS image and calculate its diagonal matrix D_N:

$$D_N = U^T C_N U \tag{6}$$

where U is an orthogonal matrix composed of eigenvectors and D_N is a diagonal matrix of C_N whose eigenvalues are arranged in descending order.

Equation (6) can be further transformed to obtain:

$$P^T \mathrm{C_N} P = I \tag{7}$$

where, in $P = UD_N^{-1/2}$, I is the unit matrix. Therefore, when P is applied to the RS image data X using the transformation $Y = PX$, the original image data can be mapped to a new space through transformation. Now, the noise in the transformed data not only has unit variance, but also the bands are not related.

(2) Standard PCA transformation of noise-whitening data:

$$C_{D-adj} = P^T C_D P \tag{8}$$

where C_D is the total covariance matrix of the image and C_{D-adj} is the transformed covariance matrix with the P transformation. Hence, the diagonalized matrix D_{D-adj} can be expressed as follows:

$$D_{D-adj} = W^T C_{\text{D-adj}} W \tag{9}$$

The two steps described above yield the MNF transformation matrix $T_{MNF} = PW$. Then the MNF-transformed RS image data are arranged in descending order of SNR, concentrating most of the information in the first few components. As the number of bands increases, the image quality is gradually degraded, thus suppressing the harmful impact of noise data on RS image quality [34], [35].

2) LBP TEXTURE FEATURE EXTRACTION

The local binary pattern (LBP), proposed by Ojala *et al.* in 1996, is an operator that describes the texture features of images based on gray scales [36]. The LBP operator has been widely used in texture classification, image recognition, and other fields because of its advantages such as fast calculation speed, gray scale invariance, and rotation invariance.

The basic LBP operator is a 3×3 rectangular window. The gray value of the center pixel is used as a threshold, and the gray values of the surrounding eight pixels are compared with it. If the gray value of the surrounding pixels is greater than or equal to the threshold, then the position of the pixel is marked as 1; otherwise, it is marked as 0. Then the eight binary values are read out in a clockwise direction from the upper left corner of the neighborhood, forming a binary string to represent the texture of the 3×3 rectangular window. Figure 7 shows the generation of a 3×3 neighborhood LBP.

After band dimension reduction, LBP feature extraction is applied to each selected single-band image after MNF dimension reduction. Figure 8 shows the LBP texture feature extraction procedure.

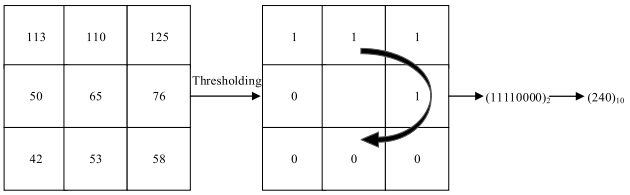


FIGURE 7. Diagram of a 3×3 neighborhood LBP computing process.

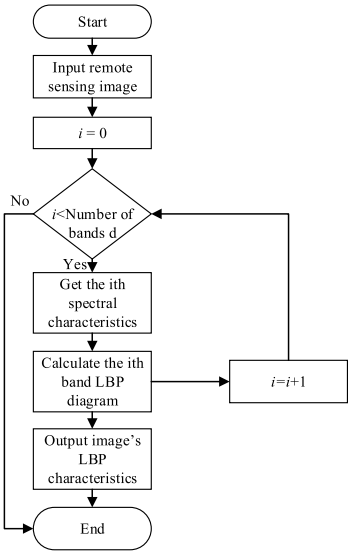


FIGURE 8. Flow diagram of LBP texture feature extraction procedure.

From Fig. 8, for a single-band image, the LBP operator of each neighborhood window is first calculated as the texture feature value of the central pixel, and the texture features of all the pixels in the image are obtained by repeatedly moving the sampling sub-window. Then the same method is used to calculate the LBP texture features of all bands in the hyperspectral RS images. The LBP values of all bands in the same pixel form a set of texture features for the pixel.

3) KELM CLASSIFICATION

After the LBP texture features have been extracted, they are used as training sample features for classification. The classifier is constructed using the RBF kernel ELM (RBF-KELM) algorithm, which uses the RBF kernel function instead of the activation function in the basic ELM algorithm to achieve further improvements in algorithm performance.

In the RBF-KELM algorithm, the kernel function $\varphi_i(\cdot), i = 1, 2, \ldots, l$ is usually a Gaussian function:

$$\varphi_i(x) = \varphi(x, \mu_i, \sigma_i) = \exp\left(\frac{\|x - \mu_i\|^2}{\sigma_i}\right) \tag{10}$$

where $\mu_i = (\mu_{i1}, \ldots, \mu_{in})^T$ is the center of the i-th core and σ_i is the extended width. Table 3 describes the RBF-KELM algorithm.

TABLE 3. ELM neural network learning algorithm.

Algorithm: RBF-KELM algorithm Input: training set $D=\{(x_i,t_i)\},i=1,2,...,n$, kernel function φ, the number of cores is l . Output: Output weight β. (1) Randomly generate kernel center μ_i and extended width σ_i, $i=1,2,...,l$; (2) Calculate the hidden layer output matrix H; (3) Calculate the output weight $\beta=H^{+}T$.

4) ANALYSIS OF THE LBP-KELM OPTIMIZATION METHOD

According to experiments, the LBP-KELM method can resolve classification instances where different objects appear to have the same spectrum or the same object appears to have different spectra, which makes the classification effect more prominent than with Ensemble-ELM. However, this method is sensitive to different datasets. To overcome this shortcoming, the following ELM optimization algorithm was developed.

C. DESIGN AND IMPLEMENTATION OF CNN-BASED ELM CLASSIFICATION OPTIMIZATION METHOD (CNN-ELM)

1) OVERALL STRATEGY AND STEPS OF CNN-ELM-BASED CLASSIFICATION

In 1962, Hubel and Wiesel discovered that their unique local interconnect network structure can effectively reduce the complexity of the feedback neural network when studying the local sensitive and directional selection of neurons in the cat's cortex, and then proposed a convolutional neural network (CNN for short) [37]. In 2006, Hinton made CNN awaken again and made great progress [38]. Subsequently, more researchers have improved the network. Among them, it is worth noting that a classic CNN architecture proposed by Krizhevsky *et al*., who showed a significant improvement in performance compared to the method before the image classification task. The overall architecture of their approach, AlexNet [39] (also known as ImageNet), is similar to LeNet-5 but has a deeper structure. Nowadays, CNN has become one of the research hotspots in many scientific fields, especially in the field of pattern classification.

There are many variants of the CNN architecture, but their basic structures are very similar. The basic architecture of CNN usually consists of three layers, a convolutional layer, a pooled layer, and a fully connected layer (as illustrated with the upper blue box in Fig. 9) [40]. The convolution layer is designed to learn the characteristic representation of the input. Each convolutional layer is usually followed by a subsampling layer. The subsampling layer is also called the pooling layer, and the pooling layer aims to achieve spatial invariance by reducing the resolution of the feature map. The lower network layer of the CNN is composed of a convolutional layer and a subsampling layer alternately, and the upper layer is a classification layer, usually called fully connected layer. The input of the classification layer is a feature image obtained by extracting features from the convolution layer and the subsampling layer, and then the input image is classified by an algorithm such as Softmax regression or SVM.

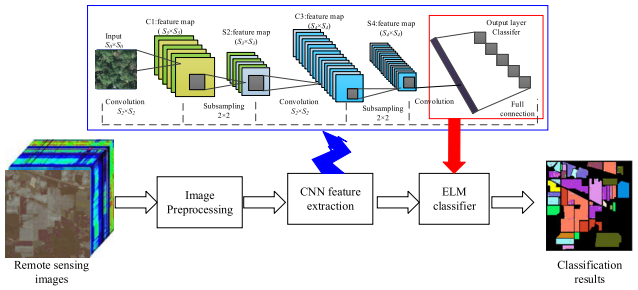


FIGURE 9. Framework and flowchart of RS image classification based on the CNN-ELM-based model.

Therefore, to improve classification accuracy and meet the classification requirements of various datasets, this study has made full use of the advantages of CNN and ELM by combining the two to construct a hyperspectral RS image classification method. The optimization focus of this method is mainly to add the depth feature extraction layer to the original ELM classification method. In this optimization method, the depth features of RS images are extracted by alternately connecting the convolutional layer and the subsampling layer in CNN. The fully connected layer in the CNN is replaced with the ELM classifier (as illustrated with red box in Fig. 9). By such optimization, the number of network parameters is greatly reduced, which significantly improves calculation efficiency. Figure 9 presents a summary flowchart of the model proposed in this paper.

Figure 9 shows that the processing flow of the classification framework includes three main steps:

(1) RS image preprocessing. When the number of RS image bands is large, to reduce computational complexity, the image is first subjected to PCA dimensionality reduction to obtain the m-dimensional features of the image. Because the CNN input data are a two-dimensional images, the output results, such as the prediction category, are directly produced by operating on the input images. For the RS image, it is necessary to predict the feature type of each pixel. To meet this requirement, sample windows are opened with the current pixel as the center. In these windows, the m-dimensional features of the pixel are respectively expanded by the N×N neighborhood, and then all the pixels are represented in image form as an input to the CNN.

(2) Feature extraction. The depth features of the RS image are extracted by alternately connecting to the convolutional layer and the sub-sampling layer of the CNN. Adding the depth feature extraction layer is the main optimization part of the original ELM classification method. The design of the feature extraction layer of this part will be highlighted in Subsection 2).

(3) Classification. The ELM algorithm is used as the classifier, and the features extracted by the CNN are input as sample features. The output weights are obtained through a large number of sample training calculations, the images

corresponding to each pixel are classified, and the classification categories are predicted. The use of ELM for classification requires manual setting of the number of hidden layer nodes L. Currently, there is no given standard for the value of L, which depends on the specific situation. This study has obtained good-quality values of L in classifying the Indian Pines and PaviaU datasets using ELM in several experiments.

2) DESIGN OF THE DEPTH FEATURE EXTRACTION LAYER

The depth feature extraction layer of the CNN-ELM method is constructed by alternately connecting two convolutional layers and two sub-sampling layers of CNN on the basis of removing the fully linked layer of CNN (as illustrated with the upper blue box in Fig. 9). The first layer of the feature extraction layer is the convolution layer, C_1. Each convolution kernel acts on the receptive field of the input image, carries out convolution operations on the image, and then obtains the M characteristic graph by nonlinear transformation of the activation function. M is not only the number of feature graphs, but also the number of convolution kernels, and it is set in the experiment according to specific conditions. Assuming that the size of the input image is $s_0 \times s_0$, the size of each characteristic graph is $s_1 \times s_1$, and the convolution kernel size is $s_2 \times s_2$, meaning that the relationship $s_1 = s_0 - s_2 + 1$ is satisfied. In general, the formula for extracting feature maps is:

$$x_j^1 = \sigma\left(w_j^1 * x + b_j^1\right), \quad j = 1, 2, \ldots, M \tag{11}$$

where x is the input image, $*$ represents the convolution operation, w_j^1, b_j^1 are the parameter and bias of the j-th convolution kernel, and x_j^1 is the j-th characteristic graph. M is the number of characteristic graphs, and $\sigma(\cdot)$ is the activation function in the convolution layer. In this study, the activation function used was a linear rectification function, ReLU: $\sigma(x) = \max(0, x)$.

A sub-sampling layer, S_2, is tightly connected to the convolution layer, and its characteristic nucleus size is set to 2×2. In the sub-sampling layer, the size of the feature graph remains the same, i.e., M, but the size of each feature graph output from the layer is reduced to $s_3 \times s_3$, where $s_3 = [s_1/2]$. The output of the S_2 layer can be expressed as:

$$x_j^2 = \sigma\left(w_j^2 down\left(x_j^1\right) + b_j^2\right) \tag{12}$$

where w_j^2 represents the parameters of the S_2 layer, b_j^2 points to the offset of the S_2 layer, and $down(\cdot)$ means the down sampling method. In this study, the selected method was maximum pooling.

The construction process and principle of the second convolution layer C_3 and the sub-sampling layer S_4 are similar to those described above.

All the weight values of the depth feature extraction layer are trained by the original CNN through the backpropagation algorithm. When the network structure converges, the weight value of the network is determined. The specific training process is as follows:

(1) The weights and biases of the networks are generated randomly.

(2) Sample x is input; after calculations by the four network layers C_1, S_2, C_3, and S_4, the classification prediction result y is finally obtained.

(3) Gradient descent training is carried out between the predicted class result y and the error of the true category label t, i.e., E, to adjust the network parameters. The adjustment formula is:

$$E = \frac{1}{2}\|t - y\|_2^2 \tag{13}$$

(4) Repeat steps 2 and 3 until the network converges.

3) IMPLEMENTATION OF CNN-ELM MODEL ON KERAS

The important contribution of this research is to implement the designed CNN-ELM classification model in the Keras framework. Compared with other deep learning frameworks such as TensorFlow and Theano, less code is needed to construct the model in Keras, and it presents a simpler structure to facilitate building the model quickly. At the same time, this framework provides enhanced model usability and extensibility, which will be beneficial to the combination of ELM and deep learning in the future.

TABLE 4. Pseudocode for implementing the CNN-ELM model.

```
# Construct a CNN model
def cnn_generate():
cnn_model = Sequential()  # construct a Sequential model
# Convolution layer
cnn_model.add(Conv2D(90, (3,3), input_shape=data_train.shape[1:]))
cnn_model.add(Activation('relu')) # Convolutional function
cnn_model.add(MaxPool2D(pool_size=(2,2))) # Subsampling layer
…# Convolution layer
…# Subsampling layer
cnn_model.add(Flatten())
# Fully connected layer
cnn_model.add(Dense(200), activation('relu'))
cnn_model.add(Dense(nb_classes),activation='softmax')
…# Model compilation
# Construct CNN middle layer output model
def hidden_layer_generate(cnn_model):
  layer_name='flatten_1'
hidden_layer_model=Model(inputs=cnn_model.input,outputs=cnn_model.get_
layer(layer_name).output) # Construct CNN middle layer
cnn_train_result=hidden_layer_model.predict(data_train) # middle layer output
return hidden_layer_model, cnn_train_result
# Construct an ELM classification model
def elm_model_generate(data_train, target_train):
  target_train_oh = np_utils.to_categorical(target_train, NUM_CLASS)
elm_model=hpelm.elm.ELM(cnn_train_result.shape[1],NUM_CLASS)
elm_model.add_neurons(ELM_HIDDEN_NEURONS, func='sigm')
elm_model.train(data_train, target_train_oh,'c') #train the model
return elm_model
```

The implementation of the CNN-ELM network model on Keras mainly included building the CNN model to obtain the feature extraction layer parameters, constructing the CNN intermediate layer output model to obtain the depth characteristics of the RS image, and constructing the ELM classification model. Table 4 gives a concrete implementation in pseudocode.

4) ANALYSIS OF THE CNN-KELM OPTIMIZATION METHOD

From the experiments described below, this study found that the CNN-ELM method has advantages such as higher precision and lower sensitivity to dataset type.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section describes the verification through experiments of the effectiveness of three kinds of classification optimization methods for RS images: Ensemble-ELM, LBP-KELM, and CNN-ELM.

A. INTRODUCTION TO THE EXPERIMENTAL DATASETS

To verify various performance indices such as classification accuracy, efficiency, and robustness for the proposed optimization algorithms, this study selected the following three kinds of hyperspectral datasets as experimental data.

1) INDIAN PINES

Indian Pines is a dataset collected by airborne visible infrared imaging spectrometer (AVIRIS) over a test area in Indiana, U.S.A., with a spatial resolution of around 20 m. The size of the image was 145×145 pixels, and it contained 224 spectral bands. In the experiment, the noise band, which was affected by air water absorption, was removed, and the tests were performed with the remaining 200 bands. The image included 16 farm objects, with two-thirds covered by crops and one-third by forest or other perennial plants. In total, the dataset contained 10,249 sample points with labels. Table 5 gives the number of different names and corresponding sample points. Figure 10 shows the false color and ground-truth data.

TABLE 5. Indian Pines farm objects and corresponding number of sample points.

Classes Index	Classes Name	Samples' number
1	Alfalfa	46
2	Corn-notill	1,428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2,455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1,265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

2) PAVIAU

The PaviaU data were collected using reflective optics system imaging spectrometer (ROSIS) sensors over the University of Pavia in northern Italy, with a spatial resolution of 1.3 m.

FIGURE 10. Indian Pines false color map and ground-truth data graph: (a) false color map; (b) ground-truth data graph.

TABLE 6. PaviaU farm objects and corresponding number of sample points.

Classes Index	Classes Name	Samples' number
1	Asphalt	6,631
2	Meadows	18,649
3	Gravel	2,099
4	Trees	3,064
5	Painted metal sheets	1,345
6	Bare Soil	5,029
7	Bitumen	1,330
8	Self-Blocking Bricks	3,682
9	Shadows	947

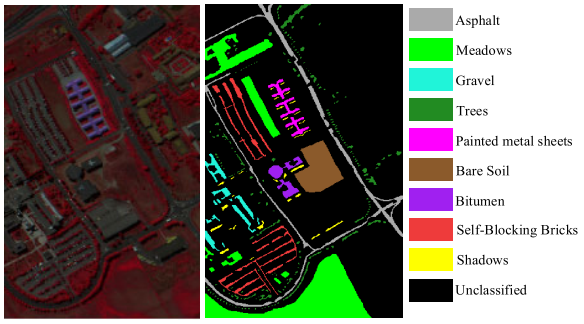


FIGURE 11. PaviaU false color map and ground-truth data graph: (a) false color map; (b) ground-truth data graph.

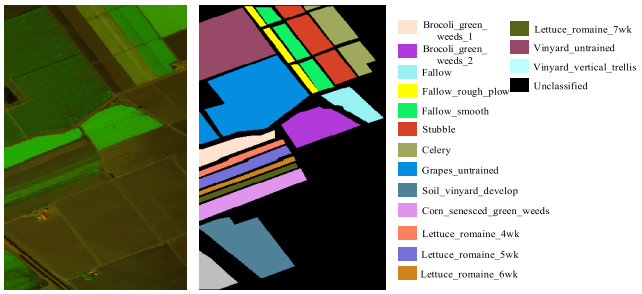
Each image had a total of 103 spectral bands, and the image size was 610×610 pixels. Some samples did not contain any information and were removed before analysis. The resulting image size was 610×340 pixels. The image included nine farm objects. The number of sample points with category labels was 42,776. Table 6 presents the number of different names and corresponding sample points. PaviaU's false color map and its ground-truth data are shown in Fig. 11.

3) SALINAS

Salinas is a high-resolution spectral dataset collected by the AVIRIS spectrometer over the Salinas Valley in the United States, with a spatial resolution of 3.7 m. The size of the image was 512×217 pixels, and it contained 224 bands. In this experiment, 20 noise bands (108–112, 154–167, 224) were removed, and the remaining 204 bands were used for classification experiments. The image contained 16 farm objects and

TABLE 7. Salinas farm objects and corresponding number of sample points.

Classes Index	Classes Name	Samples' number
1	Brocoli_green_weeds_1	2,009
2	Brocoli_green_weeds_2	3,726
3	Fallow	1,976
4	Fallow_rough_plow	1,394
5	Fallow_smooth	2,678
6	Stubble	3,959
7	Celery	3,579
8	Grapes_untrained	11,271
9	Soil_vinyard_develop	6,203
10	Corn_senesced_green_weeds	3,278
11	Lettuce_romaine_4wk	1,068
12	Lettuce_romaine_5wk	1,927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1,070
15	Vinyard_untrained	7,268
16	Vinyard_vertical_trellis	1,807



(a) (b)

FIGURE 12. Salinas false color map and ground-truth data graph: (a) false color map; (b) ground-truth data graph.

a total of 54,129 sample points with category markers. Table 7 presents the number of different names and corresponding sample points, and Fig. 12 shows the false color map and corresponding ground-truth data.

B. EXPERIMENTAL DESIGN AND RESULTS

The following experiments were performed using the three optimization algorithms (Ensemble-ELM, LBP-KELM, and CNN-ELM) with the same three datasets described above.

1) EXPERIMENTAL DESIGN AND RESULTS FOR ENSEMBLE-ELM METHOD

The experimental platform was a computer configured with an Intel® Pentium® CPU G620 @ 2.6 GHz, 4 GB memory, and equipped with MATLAB 2010 (R2010b).

(1) Indian Pines. In the ELM classification algorithm, the number of neurons L in the hidden layer must be set before classification, and therefore this study first obtained the relationship between the value of L and the ELM algorithm classification result and determined an optimal value for subsequent classification experiments. In the Indian Pines experiment, 40% of the samples were randomly selected as training data, and the remaining 60% were used for testing.

TABLE 8. Indian pines experiments: Relationship between L, the number of neurons in the hidden layers, and ELM classification results.

L (Number of neurons in hidden layers)	500	1,000	1,500	2,000	2,500	3,000	4,000	5,000
AA	49.24	53.43	59.45	60.68	65.09	65.48	69.43	71.28
OA	63.20	66.67	69.93	71.41	75.09	75.87	76.40	76.50
Kappa	0.57	0.61	0.65	0.67	0.71	0.72	0.73	0.75
Training time (s)	0.80	4.93	17.15	36.95	31.47	43.29	71.25	92.22
Testing time (s)	0.26	0.51	0.79	1.08	1.28	1.52	1.98	2.71

Note: AA means average accuracy and OA indicates overall accuracy. Kappa points to the Kappa coefficient, which is a ratio that represents the proportion of errors that are reduced by classification and completely random classification. All of these three indicators are for characterizing the classification accuracy. The same abbreviations are used in the tables bellow.

TABLE 9. Indian pines experiments: Comparison of classification results with the ELM and Ensemble-ELM algorithms (classification accuracy %).

Type	Number of training samples	Number of testing sample	ELM	Ensemble-ELM
1	19	27	67.86	85.71
2	572	856	80.51	93.47
3	332	498	45.98	53.21
4	95	142	53.15	70.63
5	194	289	88.62	92.76
6	292	438	96.35	99.77
7	12	16	70.59	88.24
8	192	286	99.30	100.00
9	8	12	33.33	58.33
10	389	583	76.54	85.62
11	982	1,473	79.09	88.93
12	238	355	51.69	64.33
13	82	123	94.31	98.37
14	506	759	92.36	98.16
15	155	231	53.02	62.93
16	38	55	33.93	83.93
	AA		69.79	82.77
	OA		77.18	86.20
	Kappa		0.74	0.84
	Training time (s)		92.06	850.00
	Testing time (s)		2.52	24.00

The activation function was sigmoid. Table 8 presents the experimental results of ELM RS image classification with different numbers of hidden layer nodes; the results are obtained from the average of 10 identical experiments.

Table 8 shows that the classification accuracy increased with the number L of hidden-layer neurons by using ELM to classify the objects in the Indian Pines dataset. However, increasing trend slowed down with further increases in L, and in the $L = 5000$ case, it basically achieved a stable value. In subsequent experiments, L was set to 5000. After several experiments, the number of base classifiers in the Ensemble-ELM algorithm in this experiment was set to 10. Table 9 lists the results of Indian Pines feature classification using the

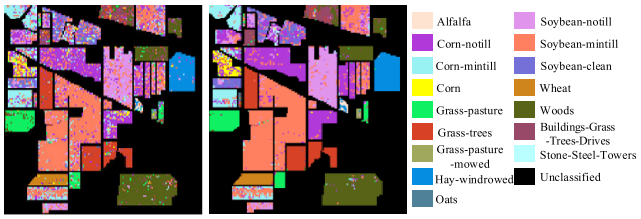


FIGURE 13. Classification results of the Indian Pines dataset with ELM and Ensemble-ELM: (a) ELM (71.73%); (b) Ensemble-ELM (84.17%).

TABLE 10. Indian pines experiments: Comparison of classification results with the ELM, Ensemble-ELM, and SVM algorithms.

Algorithm	ELM	Ensemble-ELM	SVM
AA	65.99	85.50	75.05
OA	63.25	87.77	69.18
Kappa	0.50	0.86	0.65
Training time (s)	1.36	14.10	4,350.00
Testing time (s)	3.07	25.24	1.54

TABLE 11. PaviaU experiments: Relationship between *L*, the number of nodes in hidden layers, and ELM classification results.

L (Number of neurons in hidden layers)	100	300	500	1000	1500	2000
AA	69.98	74.67	77.51	64.98	76.53	76.49
OA	57.83	67.91	72.37	62.09	76.73	76.74
Kappa	0.48	0.59	0.64	0.54	0.69	0.69
Training time (s)	0.23	0.85	1.62	6.19	19.00	41.49
Testing time (s)	0.21	0.50	0.77	1.79	2.49	3.59

ELM and Ensemble-ELM algorithms. Figure 13 shows the experimental results of classifying all the labeled samples of the Indian Pines dataset with the two algorithms.

In the case of a small sample set, a small number of samples from each category were selected as training samples, and the remaining samples were tested. Table 10 shows the results of the comparison between SVM, ELM, and Ensemble-ELM.

(2) PaviaU. In the PaviaU experiment, the number of hidden layer nodes L was also determined first by experiments. A certain proportion of samples in each category were randomly selected as training samples, and the remaining samples were used for testing. Table 11 shows the number of training and test samples. The activation function of the ELM algorithm was sigmoid. Depending on the number of hidden layer nodes L, the experimental results of the ELM classification were different, as shown in Table 11.

The analysis showed that when using ELM to classify the objects in the PaviaU case, the classification accuracy increased as the number of neurons in the hidden layer increased, but the ascending trend was very gentle, and the classification accuracy decreased at $L = 1000$ and then rose again at $L = 2000$. Essentially, the classification accuracy had achieved a stable value. The drop in classification accuracy at $L = 1000$ was caused by one experiment with a classification accuracy of only 10%, which was caused by the instability of the ELM algorithm itself.

In this experiment, L was set to 2000 and the number of base classifiers of the Ensemble-ELM algorithm to 5.

Table 12 shows the experimental results of classifying the PaviaU dataset using both ELM and Ensemble-ELM algorithms. Figure 14 shows the classification results with the ELM and Ensemble-ELM algorithms for all tagged samples in PaviaU.

TABLE 12. PaviaU experiments: Comparison of classification results with the ELM and Ensemble-ELM algorithms (classification accuracy %).

Type	Number of training samples	Number of testing samples	ELM	Ensemble-ELM
1	544	6,087	69.16	95.17
2	539	18,110	69.40	90.96
3	393	1,706	57.03	80.07
4	524	2,540	94.13	99.92
5	265	1,080	98.80	100.00
6	534	4,495	70.66	85.43
7	376	954	75.26	96.44
8	512	3,170	75.43	87.89
9	232	715	87.69	99.86
AA			77.51	92.86
OA			72.37	91.39
Kappa			0.64	0.88
Training time (s)			1.62	8.12
Testing time (s)			0.77	3.43

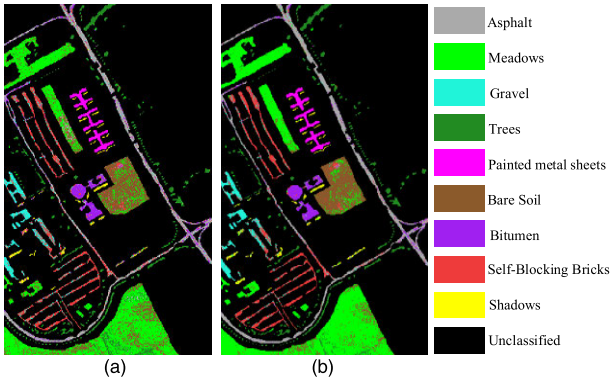


FIGURE 14. Classification results for the PaviaU dataset with ELM and Ensemble-ELM: (a) ELM (76.69%); (b) Ensemble-ELM (86.76%).

From each category, 30 samples were selected as training samples, with the rest used as test samples. Table 13 presents a comparison of the results using the SVM, ELM, and Ensemble-ELM algorithms.

(3) Salinas. In the Salinas experiment, the number of ELM hidden layer nodes L used in this experiment was determined experimentally by considering classification efficiency and accuracy. In the comparative experiment with ELM and Ensemble-ELM, 3% of the samples in each category were randomly selected as training samples, and the rest were used for testing. The activation function of the ELM algorithm

TABLE 13. PaviaU experiments: Comparison of classification results with the ELM, Ensemble-ELM, and SVM algorithms.

Algorithm	ELM	Ensemble-ELM	SVM
AA	74.12	94.36	86.96
OA	66.62	93.26	82.05
Kappa	0.58	0.91	0.77
Training time (s)	0.78	7.24	1,100.00
Testing time (s)	10.90	90.45	1.41

TABLE 14. Salinas experiments: Comparison of classification results with the ELM and Ensemble-ELM algorithms (classification accuracy %).

Type	Number of training samples	Number of testing samples	ELM	Ensemble-ELM
1	60	1949	99.28	97.69
2	112	3614	95.88	98.26
3	59	1917	73.76	95.04
4	42	1352	90.61	98.97
5	80	2598	96.81	93.88
6	119	3840	95.24	99.01
7	107	3472	98.88	99.02
8	338	10933	81.69	88.17
9	186	6017	96.84	98.07
10	98	3180	83.36	75.31
11	32	1036	81.95	90.06
12	58	1869	97.06	99.68
13	27	889	89.65	97.41
14	32	1038	84.97	92.87
15	218	7050	45.89	54.71
16	54	1753	89.73	94.35
AA			87.60	92.03
OA			84.19	88.31
Kappa			0.82	0.87
Training time (s)			2.07	52.15
Testing time (s)			2.89	52.13

was sigmoid, and the number of hidden layer nodes was set to 1000 (After the value of L reaching to 1000, the AA value of this method is getting smaller and the training time is significant increasing. Because the curve trend of the relationship between the number of neurons in the hidden layers with the classification results is exactly the same as that in the foregoing two experiments. For the space limitation, the corresponding data are note provided here). The number of base classifiers in the Ensemble-ELM algorithm was set to 20.

Table 14 shows the classification results for the Salinas features using the ELM and Ensemble-ELM algorithms. Figure 15 presents a graphical representation of the classification of all labeled samples in Salinas by the ELM and Ensemble-ELM algorithms.

From each category, 0.5% of the samples were selected as training samples, with the rest used as test samples. The number of hidden layer neurons L was set to 2000. Using the SVM, ELM, and Ensemble-ELM algorithms, the results were as shown in Table 15.

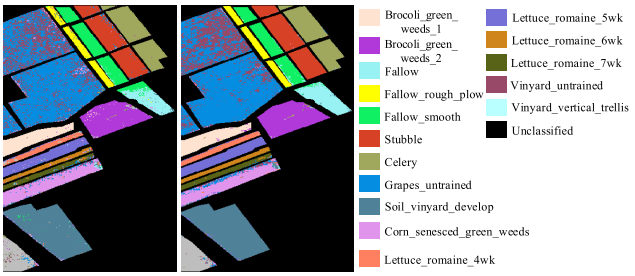


FIGURE 15. Classification results for the Salinas dataset with ELM and Ensemble-ELM: (a) ELM (84.36%); (b) Ensemble-ELM (88.46%).

TABLE 15. Salinas experiments: Comparison of classification results with the ELM, Ensemble-ELM, and SVM algorithms.

Algorithm	ELM	Ensemble-ELM	SVM
AA	72.21	91.72	90.83
OA	70.46	86.73	87.55
Kappa	0.67	0.85	0.86
Training time (s)	0.19	9.32	1825.21
Testing time (s)	5.64	106.95	2.71

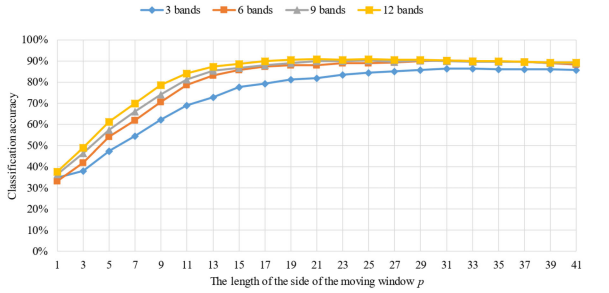


FIGURE 16. Effect of size and number of bands in the LBP shifting window on the classification accuracy of the PaviaU dataset.

2) EXPERIMENT DESIGN AND RESULTS FOR THE LBP-KELM METHOD

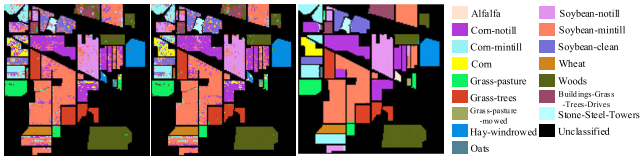
The experimental platform and the software and hardware configuration were identical to those in Subsection V.B.1). In this experiment, several parameters had to be adjusted manually: the number of bands K after MNF dimension reduction, parameters (m, r) for calculating LBP, the size of the shifting window $(p \times p)$, and the number of nuclei l in the RBF-KELM algorithm. The number of RBF-KELM nuclei in this experiment was set to 5. For other parameters, the PaviaU data were taken as an example to introduce the process of adjusting and determining parameter value. When (m, r) were fixed as (8, 2), the classification precision was affected by the LBP shifting window size $(p \times p)$ and the number of bands k (see Fig. 16). It is clear from Fig. 16 that when the number of bands $k > 6$ and the window size is 21×21, the classification accuracy reaches the maximum value. It can be concluded that when the number of bands $k = 10$ and the window size is 21×21, the relationship between LBP-KELM classification accuracy and the parameters of the LBP operator (m, r) is

TABLE 16. Overall classification accuracy of the PaviaU dataset using the LBP-KELM algorithm with different parameters (*m, r*) (%).

m / r	4	6	8	10
1	87.28	89.41	89.33	90.30
2	86.67	89.18	90.55	90.21
3	85.55	90.85	90.26	90.80

TABLE 17. Optimal parameter values in the classification experiments.

Parameters	PaviaU	Indian Pines	Salinas
l	5	5	5
k	10	12	10
(m,r)	(6,3)	(8,2)	(8,2)
p×p	21×21	21×21	21×21



(a) (b) (c)

FIGURE 17. Classification results for the Indian Pines dataset with KELM, SVM, and LBP-KELM: (a) KELM (79.09%); (b) SVM (81.43%); (c) LBP-KELM (98.48%).

as shown in Table 16. The classification accuracy is highest when $(m, r) = (6, 3)$. When $m \geq 6$, the classification accuracy tends to stabilize and is no longer sensitive to different R values. Because adjacent pixels in space likely belong to the same category, R should be set to a smaller value. The features of LBP have $M \times (m-1) + 3$ dimensions. The larger the value of M, the higher will be the computational complexity. Therefore, considering classification precision and computational complexity, in the PaviaU case, $(m, r) = (6, 3)$ gave the best results.

In the classification experiments with the Indian Pines and Salinas datasets, the parameter tuning process of the LBP-KELM algorithm was the same as with PaviaU. In the classification experiments, the final set of optimal parameter values were as shown in Table 17.

(1) Indian Pines. The experiment randomly selected some samples in each category as training samples, and the remaining samples were used to test classification accuracy. The parameter settings are shown in Table 17. Table 18 presents a list of test set experimental results after classification of Indian Pines features by KELM, SVM, and LBP-KELM. Figure 17 presents a graphical representation of the classification of all the Indian Pines labeled samples by the three methods.

(2) PaviaU. This experiment randomly selected some samples in each category as training samples, and the remaining samples were used to test classification accuracy. Table 19 gives a list of test set experimental results from KELM, SVM, and LBP-KELM classification of PaviaU features. Figure 18 provides a graphical representation of the classification of all PaviaU labeled samples by the three methods.

(3) Salinas. This experiment randomly selected 1% of the samples in each category as training samples, and the remaining samples were used to test classification accuracy. Table 20 gives a list of test set experimental results after classification of Salinas features by KELM, SVM, and LBP-KELM. Figure 19 provides a graphical representation of

TABLE 18. Indian pines experiments: Comparison of classification results with the KELM, SVM, and LBP-KELM algorithms (%).

Index	Number of training samples	Number of testing samples	KELM	SVM	LBP-KELM
1	5	41	4.88	17.07	97.56
2	143	1,285	75.10	73.93	98.13
3	83	747	52.07	57.03	99.33
4	24	213	34.27	55.87	95.31
5	48	435	86.67	86.90	99.08
6	73	657	98.33	96.19	99.09
7	3	25	44.00	72.00	100.00
8	48	430	99.30	98.84	100.00
9	2	18	0.00	38.89	100.00
10	97	875	65.03	70.17	94.51
11	246	2,209	84.83	86.56	99.05
12	59	534	60.67	66.10	96.07
13	21	184	98.37	99.46	100.00
14	127	1,138	97.36	96.49	99.56
15	39	347	51.01	56.48	99.14
16	9	84	72.62	95.24	96.43
	AA		64.03	72.95	98.33
	OA		77.90	80.22	98.36
	Kappa		0.74	0.77	0.98
	Training time (s)		0.11	2,6600.00	0.22
	Testing time (s)		0.61	8.60	1.51

TABLE 19. PaviaU experiments: Comparison of classification results with the KELM, SVM, and LBP-KELM algorithms (%).

Index	Number of training Samples	Number of testing Samples	KELM	SVM	LBP-KELM
1	66	6565	87.52	85.80	84.30
2	186	18463	98.79	95.50	99.02
3	21	2078	66.27	67.37	81.28
4	31	3033	88.00	82.43	60.70
5	13	1332	98.42	99.32	92.04
6	50	4979	68.41	80.10	99.50
7	13	1317	46.62	75.70	84.43
8	37	3645	83.90	78.27	94.38
9	9	938	99.57	99.89	42.22
	AA		81.94	84.93	81.98
	OA		88.20	87.99	90.85
	Kappa		0.84	0.84	0.88
	Training time (s)		0.02	2,170.00	0.13
	Testing time (s)		0.91	1.90	1.59

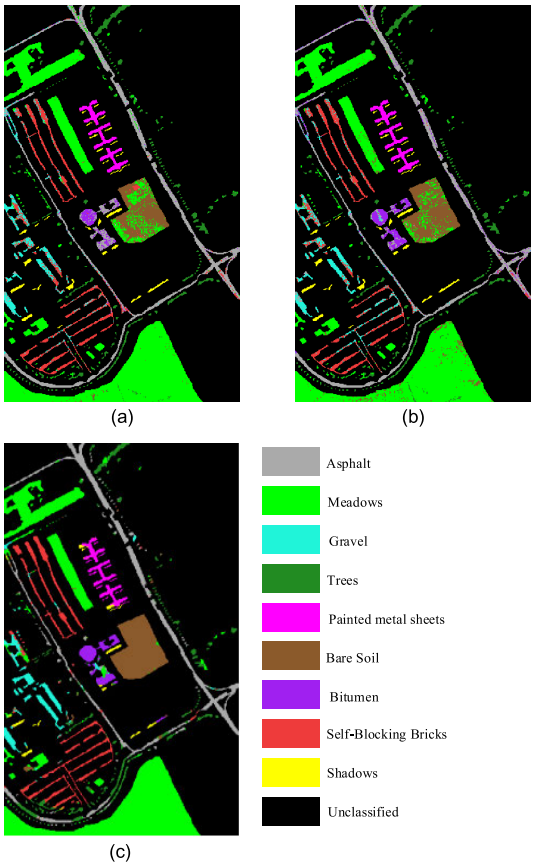


FIGURE 18. Classification results of the PaviaU dataset with KELM, SVM, and LBP-KELM:(a) KELM (88.29%); (b) SVM (87.22%); (c) LBP-KELM (90.94%).

the classification of all labeled samples in the Salinas dataset by the three methods.

3) EXPERIMENTAL DESIGN AND RESULTS FOR CNN-ELM METHOD

This experimental platform was a computer configured with an Intel® Core™ i7-6498DU@ 2.50 GHz, 8 GB memory, and the operating system is Ubuntu equipped with Keras package. Due to the fact the different experimental platforms have different experimental environments, the comparison of the training time and test time in this experiment has no comparability and are unnecessary. Therefore, the following experiments are primarily intended to present the comparison of the classification accuracy. Table 21 lists the parameters of the CNN-ELM classification method, where the loss function is *categorical_crossentropy*.

(1) Indian Pines. In the Indian Pines experiment, as in the experiment described in Subsection V.B.1), 40% of the samples were randomly selected as training samples, and the remaining 60% were used for testing. Table 22 shows the experimental results of the classification of Indian Pines RS images by the ELM and CNN-ELM algorithms. Figure 20 provides a graphical representation of the classification of all labeled samples in the Indian Pines RS images by the two algorithms.

TABLE 20. Salinas experiments: Comparison of classification results with the KELM, SVM, and LBP-KELM algorithms (%).

Index	Number of training samples	Number of testing samples	KELM	SVM	LBP-KELM
1	20	1989	98.99	98.84	99.85
2	37	3689	99.24	95.66	94.39
3	20	1956	49.59	91.46	95.91
4	14	1380	97.39	97.25	98.33
5	27	2651	99.25	98.23	97.93
6	40	3919	99.74	99.74	95.92
7	36	3543	99.69	99.55	98.02
8	113	11158	88.86	83.62	98.22
9	62	6141	99.90	99.67	98.03
10	33	3245	87.92	89.55	96.36
11	11	1057	90.73	93.57	100.00
12	19	1908	99.74	98.79	98.79
13	9	907	99.01	98.90	89.75
14	11	1059	87.54	88.39	97.73
15	73	7195	55.80	63.68	96.32
16	18	1789	98.27	94.19	93.52
	AA		90.73	93.19	96.82
	OA		88.42	89.57	97.09
	Kappa		0.87	0.88	0.97
	Training time (s)		0.03	6148.68	0.04
	Testing time (s)		1.97	4.08	3.96

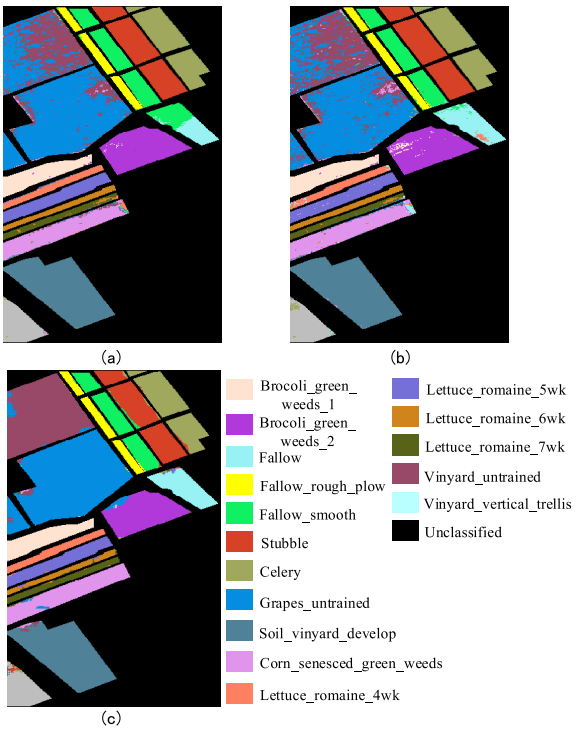


FIGURE 19. Classification results of the Salinas dataset with KELM, SVM, and LBP-KELM: (a) KELM (88.46%); (b) SVM (89.64%); (c) LBP-KELM (97.12%).

(2) PaviaU. In the PaviaU experiment, 5% of the samples in each class were randomly selected as training samples, and the rest were used for testing. Table 23 shows the test sample

TABLE 21. Parameter settings for CNN-ELM.

Parameters	Indian Pines	PaviaU	Salinas
Number of PCA principal components	30	30	30
Image sampling window size	5×5	5×5	5×5
Number of convolution kernels	90	90	90
Size of convolution kernel	3×3	3×3	3×3
Number of ELM hidden neurons	300	200	200
Learning step size	0.01	0.01	0.01

TABLE 22. Indian pines experiments: Comparison of classification results with the ELM and CNN-ELM algorithms (%).

Classes Index	ELM	CNN-ELM
1	67.86	100.00
2	80.51	92.36
3	45.98	87.81
4	53.15	92.09
5	88.62	96.63
6	96.35	98.87
7	70.59	94.44
8	99.30	99.65
9	33.33	92.31
10	76.54	90.56
11	79.09	91.87
12	51.69	91.69
13	94.31	99.19
14	92.36	97.14
15	53.02	96.12
16	33.93	84.85
AA	69.79	93.50
OA	77.18	93.48
Kappa	0.74	0.92

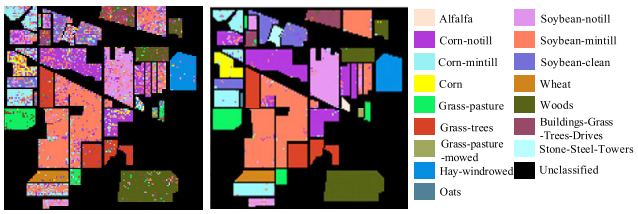


FIGURE 20. Classification results of the Salinas dataset with ELM and CNN-ELM: (a) ELM (71.73%); (b) CNN-ELM (96.08%).

classification results. Figure 21 shows the classification effect of ELM and CNN-ELM algorithms on all labeled samples in the PaviaU images.

(3) Salinas. In the Salinas experiment, 3% of the samples were randomly selected as training samples, and the rest were used for testing. Table 24 shows the test sample classification results. Figure 22 shows the classification effect of the ELM and CNN-ELM algorithms on all labeled samples in the Salinas images.

TABLE 23. PaviaU experiments: Comparison of classification results with the ELM and CNN-ELM algorithms (%).

Classes Index	ELM	CNN-ELM
1	76.89	97.30
2	93.36	97.51
3	58.12	84.80
4	77.35	99.02
5	69.93	100.00
6	53.00	95.78
7	65.40	84.82
8	74.18	86.81
9	6.79	98.79
AA	63.89	95.55
OA	78.01	95.52
Kappa	0.70	0.94

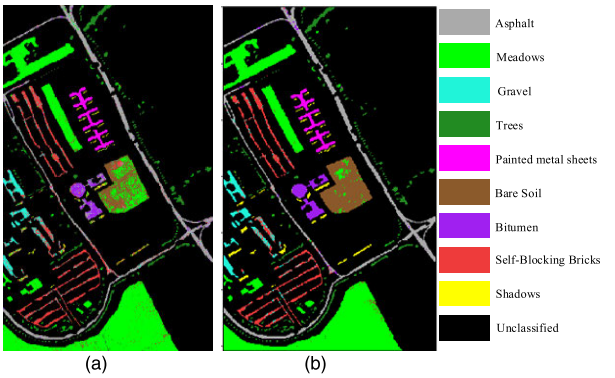


FIGURE 21. Classification results of the PaviaU dataset with ELM and CNN-ELM: (a) ELM (79.05%); (b) CNN-ELM (95.74%).

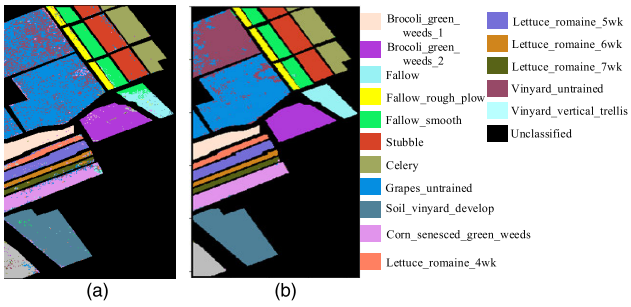


FIGURE 22. Classification results of the Salinas dataset with ELM and CNN-ELM: (a) ELM (84.36%); (b) CNN-ELM (93.48%).

C. ANALYSIS OF THE EXPERIMENTAL RESULTS

By analyzing the classification results on the three datasets from the experiments described above, the following conclusions can be drawn:

(1) The Ensemble-ELM algorithm has advantages both in classification accuracy and efficiency. The robustness of the algorithm is better, and the RS image classification method based on ELM is optimized. In particular, the classification result on the PaviaU dataset was better, with OA improved by 19%, which was very impressive. Compared with ELM,

TABLE 24. Salinas experiments: Comparison of classification results with the ELM and CNN-ELM algorithms (%).

Classes Index	ELM	CNN-ELM
1	99.28	100.00
2	95.88	99.97
3	73.76	99.38
4	90.61	99.63
5	96.81	98.51
6	95.24	100.00
7	98.88	100.00
8	81.69	92.89
9	96.84	99.32
10	83.36	99.08
11	81.95	99.61
12	97.06	99.63
13	89.65	99.89
14	84.97	98.65
15	45.89	86.37
16	89.73	99.94
AA	87.60	98.30
OA	84.19	96.37
Kappa	0.82	0.96

TABLE 25. Comparison of these three ELM optimization methods.

Characteristics	Ensemble-ELM	LBP-KELM	CNN-ELM
Robustness Enhanced	Yes	Yes	Yes
Classification accuracy	Improved	Further improved	Further improved
Classification effect	Not ideal	Satisfactory	Satisfactory
Sensitive to datasets		Yes	NO

the RS image classification effect with Ensemble-ELM was better. The number of noise points was greatly reduced, but the effect was still not ideal. Compared with SVM, Ensemble-ELM still has obvious advantages in classification speed, which satisfies the demands of real-world applications.

(2) The LBP-KELM method makes full use of the spatial texture characteristics of RS images, overcoming classification issues such as different objects that appear to have the same spectrum and the same object appearing to have different spectra, reducing noise points in the classification chart significantly. The image is smoother, and the optimization effect is more prominent than with Ensemble-ELM. LBP-KELM produced good results when classifying of the Indian Pines and Salinas datasets, but the classification effect on PaviaU was not obvious. OA was improved by only about 2.5%; this indicates that the classification effect is easily influenced by the dataset used, which makes the method unsuitable for classifying RS images with slender regions.

(3) The CNN-ELM method gave the ideal classification results for the three kinds of datasets: the classification precision was greater than 93%, which shows that CNN-ELM has an outstanding effect on ELM-based RS image classifications and has low sensitivity to dataset characteristics, which makes the method suitable for a variety of datasets.

Based on the above analysis, we can get a concise comparison to illustrate the characteristics of these three optimization methods (Table 25).

VI. CONCLUSION AND FUTURE WORK

In this study, the ELM algorithm was used to classify hyperspectral RS images. The study mainly aimed to solve the problems of the ELM algorithm in classification, such as instability, poor robustness, and low classification accuracy, from three aspects: integrated learning, using image texture features, and deep learning. Three kinds of RS image classification optimization methods based on ELM have been proposed. Among them, the Ensemble-ELM classification method was proposed to optimize the original ELM classification algorithm to improve its robustness and classification accuracy, but the classification effect was still not ideal at some points in the classification diagram. To overcome this, the second optimization method, LBP-KELM, further improved the classification effect. Because the spatial texture features of RS images were fully utilized, the noise points in the classification map were significantly reduced, the edge categories were more accurately classified, and the classification efficiency was higher than with Ensemble-ELM. However, the disadvantage was that LBP-KELM was not suitable for a variety of datasets. To solve this problem, the third method, CNN-ELM, obtained ideal classification results for the three kinds of datasets; this method had a good classification effect and was applicable to a variety of datasets. According to the experiments, all the three optimization methods can achieve a balance between classification accuracy and efficiency, i.e., they can maintain the advantage of ELM algorithm in classification efficiency and speed while have better classification accuracy. Among the three methods, it was the optimization method with the best overall effect.

However, there is still room for optimization in the proposed methods and for further improvements to the precision and speed of RS image classification. Such optimization efforts include parallelization of the Ensemble-ELM method and optimization of the stochastic parameters of the ELM algorithm. The authors will investigate these subjects in their future work.

REFERENCES

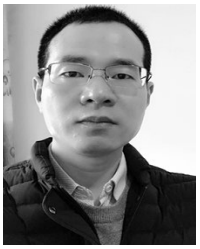
[1] C. L. Fan, "Research on hyperspectral remote sensing imagery classification algorithms based on decision tree," M.S. thesis, Yanshan Univ., Qinhuangdao, China, 2014.

[2] C. Zhao and L. Qiao, "Classification of hyperspectral remote sensing image using improved LS-SVM," *Appl. Sci. Technol.*, vol. 35, no. 1, pp. 44–47, Jan. 2008.

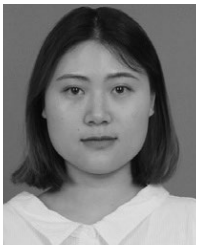
[3] Z. L. Lin and L. M. Yan, "Decision tree classification of hyperspectral remote sensing imagery based on independent component analysis," *J. Comput. Biol.*, vol. 32, no. 2, pp. 524–527, Feb. 2012.

[4] X. P. Jia, "Simplified maximum likelihood classification for hyperspectral data in cluster space," in *Proc. IGARSS*, Toronto, ON, Canada, 2002, pp. 2578–2580.

[5] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, May 2010.
[6] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
[7] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.
[8] X. H. Li and Y. Zheng, "Study on the extraction of remote sensing information of land use/cover based on optimum band combination," *J. Anhui Agricult. Sci.*, vol. 37, no. 14, pp. 6696–6699, Feb. 2009.
[9] D. C. Feng, G. Chen, W. Y. Du, X. Y. Wu, and K. L. Xiao, "Remote sensing image classification based on minimum distance method," *J. North China Inst. Aerosp. Eng.*, vol. 22, no. 3, pp. 1–2 and 5, Jun. 2012.
[10] L. Chen, X. Liu, and Y. Zhang, "A study of image classification based on MLC combined with spectral angle," *Eng. Surveying Mapping*, vol. 16, no. 3, pp. 40–42, Jun. 2007.
[11] P. F. Zhang, X. Q. Deng, and G. Q. Liu, "A comparative study of the remote sensing land use classification model for the mountain watershed area in Yunnan province," *Remote Sens. Land Resour.*, vol. 19, no. 1, pp. 89–93, Mar. 2007.
[12] J. Zhang and G. M. Foody, "A fuzzy classification of sub-urban land cover from remotely sensed imagery," *Int. J. Remote Sens.*, vol. 19, no. 14, pp. 2721–2738, Sep. 1998.
[13] Y. Q. Yang and X. R. Chai, "The research on remote sensing image classification based on artificial neural network," *J. Shanxi Normal Univ. Natural Sci. Ed.*, vol. 31, no. 1, pp. 94–98, Mar. 2017.
[14] X. C. Duan, "A remote sensing study of land use classification based on back-propagation artificial neural network," M.S. thesis, China Univ. Geosci., Beijing, China, 2008.
[15] M. Borhani and H. Ghassemian, "Kernel multivariate spectral–spatial analysis of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2418–2426, Jun. 2015.
[16] L. Z. Wang, J. B. Zhang, P. Liu, K.-K. R. Choo, and F. Huang, "Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification," *Soft Comput.*, vol. 21, no. 1, pp. 213–221, Jul. 2017.
[17] P. Liu, K. K. R. Choo, L. Wang, and F. Huang, "SVM or deep learning? A comparative study on remote sensing image classification," *Soft Comput.*, vol. 21, no. 23, pp. 7053–7065, Dec. 2017.
[18] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.
[19] G. Huang, G. B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.
[20] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
[21] M. Pal, "Extreme-learning-machine-based land cover classification," *Int. J. Remote Sens.*, vol. 30, no. 14, pp. 3835–3841, Jul. 2009.
[22] A. Samat, P. Du, J. Y. Li, L. Cheng, and S. Liu, "E2LMs: Ensemble extreme learning machines for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, Apr. 2014.
[23] M. Han and B. Liu, "Ensemble of extreme learning machine for remote sensing image classification," *Neurocomputing*, vol. 149, pp. 65–70, Feb. 2015.
[24] C. Chen, W. Li, H. Su, and K. Liu, "Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine," *Remote Sens.*, vol. 6, no. 6, pp. 5795–5814, Jun. 2014.
[25] M. Pal, A. E. Maxwell, and T. A. Warner, "Kernel-based extreme learning machine for remote-sensing image classification," *Remote Sens. Lett.*, vol. 4, no. 9, pp. 853–862, Jun. 2013.
[26] Y. Zhou, J. Peng, and C. L. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2351–2360, Jun. 2015.
[27] H. Su, S. Tian, Y. Sheng, C. Chen, M. Najafian, and Y. Cai, "Optimized extreme learning machine for urban land cover classification using hyperspectral imagery," *Frontiers Earth Sci.*, vol. 11, no. 4, pp. 765–773, Dec. 2017.
[28] Q. Lv, X. Niu, Y. Dou, J. Xu, and Y. Lei, "Classification of hyperspectral remote sensing image using hierarchical local-receptive-field-based extreme learning machine," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 434–438, Mar. 2016.
[29] F. Cao, Z. Yang, M. Jiang, W.-K. Ling, and J. Ren, "Linear vs. nonlinear extreme learning machine for spectral-spatial classification of hyperspectral images," *Sensors*, vol. 17, no. 11, p. 2603, Nov. 2017.
[30] F. Lv and M. Han, "Hyperspectral remote sensing image classification based on deep extreme learning machine," *J. Dalian Univ. Technol.*, vol. 58, no. 2, pp. 166–173, Mar. 2018.
[31] F. Lv, M. Han, and T. Qiu, "Remote sensing image classification based on ensemble extreme learning machine with stacked autoencoder," *IEEE Access*, vol. 5, pp. 9021–9031, May 2017.
[32] B. Gendron and T. G. Crainic, "Parallel branch-and-branch algorithms: Survey and synthesis," *Oper Res.*, vol. 42, no. 6, pp. 1042–1066, Nov./Dec. 1994.
[33] C. Chen, "Cloud classification of satellite imagery based on ELM and SVM," M.S. thesis, Nanchang Hangkong Univ., Nanchang, China, 2014.
[34] Y. Q. Zhu, Z. H. Sun, and X. J. Ji, "Incremental updating algorithm based on frequent pattern tree for mining association rules," *Chin. J. Comput.*, vol. 26, no. 1, pp. 91–96, Jan. 2003.
[35] B. Liu, *Web Data Mining*. Beijing, China: Tsinghua Univ. Press, 2013.
[36] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.
[37] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, Jan. 1962.
[38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[39] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, vol. 25, no. 2, pp. 1097–1105.
[40] Y. D. Li, Z. B. Hao, and H. Lei, "Survey of convolutional neural network," *J. Comput. Appl.*, vol. 36, no. 9, pp. 2508–2515, 2565, Sep. 2016.

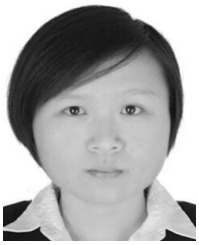


FANG HUANG received the B.S. degree in surveying and mapping engineering from the Taiyuan University of Technology, Taiyuan, China, in 2002, the M.S. degree in photogrammetry and remote sensing from Beijing Jiaotong University, Beijing, China, in 2005, and the doctor's degree in high performance geo-computation from the Institute of Remote Sensing Applications, Chinese Academy of Science, in 2008. He is currently an Associate Professor with the School of Recourses and Environment, University of Electronic Science and Technology of China. His current research interests include putting the cut-edge computing techniques, e.g., the parallel computing, grid computing, cloud computing, and even heterogeneous computing into RS/GIS algorithms' designing and implementation.

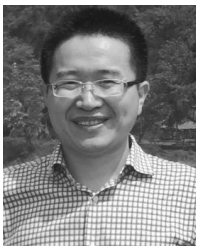


JUN LU received the B.S. degree in geographic information science from Liaoning Normal University. She is currently pursuing the master's degree in cartography and geographical information system with the School of Recourses and Environment, University of Electronic Science and Technology of China. Her current research interests include the parallel computing and real-time 3D modeling construction.

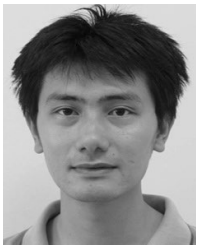
JIAN TAO received the B.S. degree in space physics from the University of Science and Technology of China, in 2000, and the Ph.D. degree in computational astrophysics from Washington University, St. Louis, in 2008. He is currently a Research Scientist with the Texas A&M Engineering Experiment Station (TEES). His research interests include high performance computing, computational framework, data analytics, machine learning, numerical modeling, and cyber infrastructure.



LI LI received the master's degree in electronics and communication engineering from the School of Recourses and Environment, University of Electronic Science and Technology of China, in 2018, and the B.S. degree in computer science and technology from Shangdong Normal University. She is currently an Engineer with Hikvision Digital Technology Co. Ltd. Her current research interest includes parallel computing.



XICHENG TAN received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, and the B.Sc. degree in surveying engineering from the Taiyuan University of Science and Technology, China, in 2007 and 2002, respectively. He is currently an Associate Professor of geographic information science with the School of Remote Sensing and Information Engineering, Wuhan University. His research interests include geospatial Web services, distributed computing, cloud computing, parallel computing, and 3D GIS.



PENG LIU received the M.S. and Ph.D. degrees in signal processing from the Chinese Academic of Science (CAS), in 2004 and 2009, respectively, where he is currently an Associate Professor with the Aerospace Information Research Institute (AIR), CAS. His research interests include big data, sparse representation, compressive sensing, deep learning, and their applications to remote sensing data processing.