Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 9150482, 20 pages
https://doi.org/10.1155/2023/9150482

*Research Article*

# Optimization Strategy of a Stacked Autoencoder and Deep Belief Network in a Hyperspectral Remote-Sensing Image Classification Model

**Xiaoai Dai,[1] Junying Cheng,[1] Shouheng Guo,[1] Chengchen Wang,[1] Ge Qu,[1] Wenxin Liu,[1] Weile Li,[2] Heng Lu,[3] Youlin Wang,[4] Binyang Zeng,[5] Yunjie Peng,[6] and Shuneng Liang[7]**

[1]*School of Earth Science Chengdu University of Technology, Chengdu 610059, China*
[2]*State Key Laboratory of Geohazard Prevention and Geoenvironment Protection, Chengdu University of Technology, Chengdu 610059, China*
[3]*College of Hydraulic and Hydroelectric Engineering, Sichuan University, Chengdu 610065, China*
[4]*Northwest Engineering Corporation Limited, Xi'an 710065, China*
[5]*Southwest Branch of China Petroleum Engineering Construction Co. Ltd, Chengdu 610095, China*
[6]*GEOVIS Wisdom Technology Co. Ltd, Qingdao 266000, China*
[7]*Land Satellite Remote Sensing Application Center, Ministry of Natural Resources of China, Beijing 100048, China*

Correspondence should be addressed to Shuneng Liang; liangsn83@163.com

Received 21 August 2022; Revised 8 October 2022; Accepted 5 April 2023; Published 28 April 2023

Academic Editor: Jinchang Ren

Improvements in hyperspectral image technology, diversification methods, and cost reductions have increased the convenience of hyperspectral data acquisitions. However, because of their multiband and multiredundant characteristics, hyperspectral data processing is still complex. Two feature extraction algorithms, the autoencoder (AE) and restricted Boltzmann machine (RBM), were used to optimize the classification model parameters. The optimal classification model was obtained by comparing a stacked autoencoder (SAE) and a deep belief network (DBN). Finally, the SAE was further optimized by adding sparse representation constraints and GPU parallel computation to improve classification accuracy and speed. The research results show that the SAE enhanced by deep learning is superior to the traditional feature extraction algorithm. The optimal classification model based on deep learning, namely, the stacked sparse autoencoder, achieved 93.41% and 94.92% classification accuracy using two experimental datasets. The use of parallel computing increased the model's training speed by more than seven times, solving the model's lengthy training time limitation.

## 1. Introduction

Hyperspectral-imaging spectral sensors, which are now carried out on a variety of platforms, such as satellites, aerospace aircraft, uncrewed aerial vehicles, and ground vehicles, collect rich surface reflectance information [1–3]. This kind of remote-sensing imaging technology with tens or hundreds of spectral bands is used in geological and mineral detection, environmental investigations [4–7], vegetation monitoring [8–10], and marine research [11]. Correction and compression of hyperspectral image data and object detection and feature classification are important research areas of hyperspectral remote sensing [12, 13]. Supervised classification algorithms commonly used for low-dimensional space perform poorly in the classification of hyperspectral data due to the Hughes phenomenon [14, 15]. For classification purposes, it is crucial to retain valuable information from high-dimensional data while reducing their dimensionality [16–18]. An urgent issue is reducing the dimensionality of the data while using a reasonable feature extraction method to derive the nonlinear structure information from the image pixels that are more suitable for classification [19, 20].

The pixel features that are more important for classification can be obtained by unsupervised classification. Commonly used hyperspectral data feature extraction methods include linear and nonlinear dimensionality reduction approaches. The main linear dimensionality reduction methods used in hyperspectral image dimensionality reduction are independent component analysis (ICA) [21, 22], principal component analysis (PCA) [23–25], linear discriminant analysis (LDA) [22, 26], and local feature analysis (LFA) [27], among others. However, traditional linear dimensionality reduction methods cannot explore the nonlinear features inside the hyperspectral data, leading to lower final classification accuracy. Therefore, there have been fewer studies on hyperspectral feature extraction algorithms for linear dimensionality reduction algorithms in recent years. The commonly used nonlinear dimensionality reduction methods are based on kernel functions and eigenvalues. The kernel function-based methods include methods such as kernel discriminant analysis (KDA) [28–30] and kernel principal component analysis (KPCA) [31].

Nonlinear dimensionality reduction methods are used more frequently in remote sensing due to the rapid advancement of deep learning theory and technology [32]. Due to their advantages, deep learning theory is more regularly applied to the feature extraction and classification of hyperspectral image data [33–36]. Deep learning is particularly suitable for feature extraction of high-dimensional nonlinear data [37, 38]. Numerous studies have used deep learning methods combined with different classifiers to achieve better classification results of hyperspectral remote-sensing images [39–42], including stacked autoencoders (SAEs), deep belief networks (DBNs), and convolutional neural networks (CNN) [43]. Deep learning algorithms combined with spatial background information can extract high-quality spectral and spatial information and achieve high classification accuracy with a small number of training samples and simple classifiers [44, 45]. Multiscale deep learning can be combined to classify hyperspectral images [46]. For example, a new hyperspectral image classification model based on deep understanding is constructed by combining stacked autoencoders (SAEs) and deep convolutional neural networks [47], and a spatial pyramidal pool is used for pooling deep convolutional neural networks, obtaining excellent classification performance [48, 49]. The stacked autoencoder (SAE), which has the advantage of better data dimensionality reduction, is used in the process of feature extraction of hyperspectral remote sensing, reducing processing complexity, and thus improving the efficiency of data abstraction and the accuracy of data classification [50]. Moreover, combined with the classification advantages of the CNN [51, 52], a fusion network for image classification can be constructed based on an SAE optimization, improving classification performance compared to traditional data processing [53, 54]. The semisupervised classification algorithm based on multilabeled samples and deep learning [55], with labels from both the nearest domain information and training samples [56, 57], and nonlabeled samples obtained from self-teaching learning, yields an effective semisupervised hyperspectral image classification method [58, 59]. Numerous classification experiments based on deep learning algorithms on a variety of hyperspectral data found that deep learning algorithms are the optimal classification algorithms in most cases [60–66].

This paper's objective was to apply deep learning theory to hyperspectral image classification, investigate the hyperspectral image classification model combined with a deep learning algorithm, and obtain a preferred method to improve classification accuracy by resolving the challenges of the Hughes phenomenon and of extracting the nonlinear features from within image elements. To achieve the objectives, we examined the development and testing of the best classification models (autoencoders (AEs) and restricted Boltzmann machines (RBMs)) based on deep learning and verified the applicability of deep learning models in the classification of hyperspectral images. We obtained the final classification model by connecting these two feature extraction algorithms and classifiers suitable for hyperspectral image classification by analyzing model building blocks. Through the experiments, we analyzed the effects of the different number of hidden layer neurons, the additional number of hidden layers, different classifiers, and other factors on the model performance and obtained an optimal classification model with optimal parameters. Finally, we proposed optimization strategies to improve classification accuracy and speed.

## 2. Methods

*2.1. Research Overview.* This paper is mainly based on hyperspectral image data, introduces the deep learning algorithm, and discusses the SAE and DBN models. Two feature extraction algorithms and classifiers suitable for hyperspectral image classification were connected to obtain the final classification model. Two sets of experimental data were used to verify the two models. Through experiments, we analyzed the influence of different hidden layer neurons, hidden layers, classifiers, and other factors on the model's performance and obtained an optimal classification model with optimal parameters. Finally, an optimization strategy was proposed to optimize and revise the model regarding classification accuracy and speed. The workflow of the process is shown in Figure 1.

### *2.2. Stacked Autoencoder (SAE)*

*2.2.1. Autoencoder (AE).* The autoencoder (AE) consists of a feedforward neural network with an input layer, a hidden layer, and an output layer (Figure 2). The AE assumes approximate equality between the decoder output features and the input features for training, allowing a large amount of unlabeled training sample data to be applied to the model's training process. As a result, overfitting and local extremes brought on by too many parameters and insufficient labeled training samples are avoided.

The training process starts with first mapping $x \in R^{n1}$ ($n1$ denotes the dimensionality of the input data, i.e., the

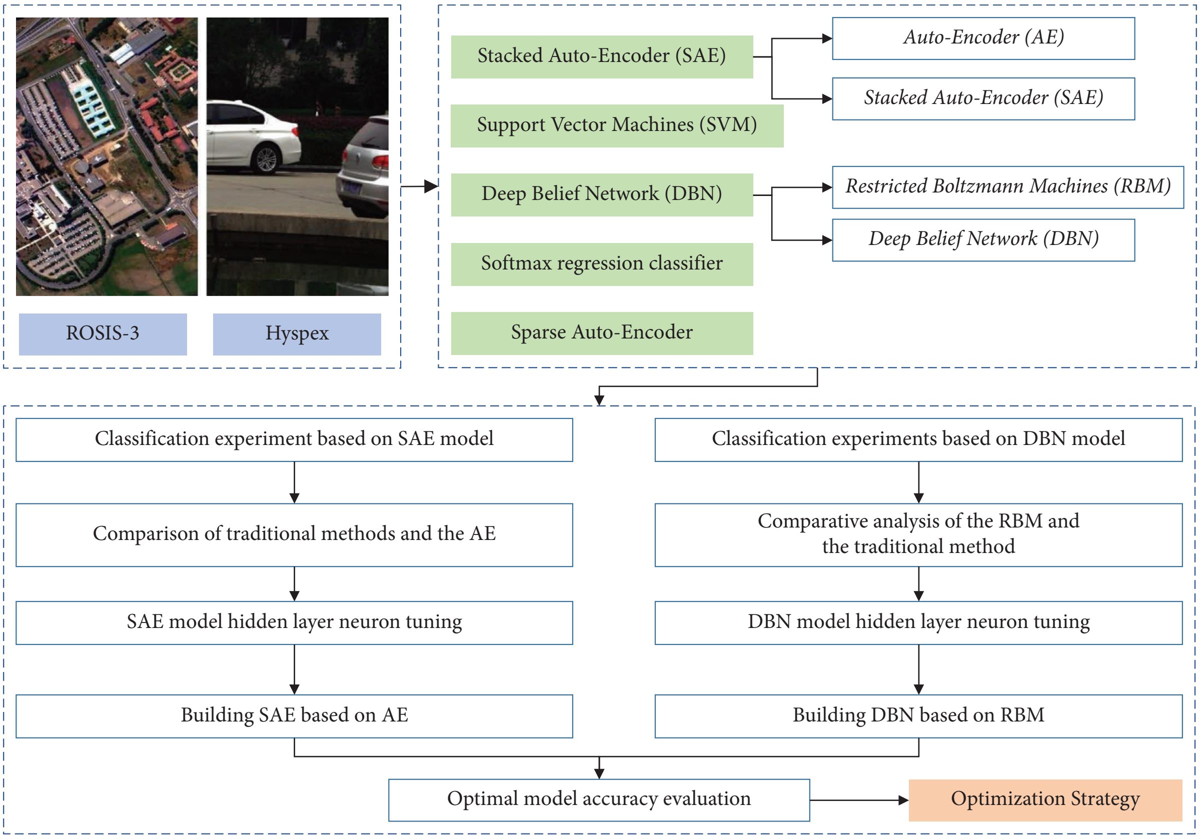

ROSIS-3
Hyspex
Stacked Auto-Encoder (SAE)
Support Vector Machines (SVM)
Deep Belief Network (DBN)
Softmax regression classifier
Sparse Auto-Encoder
Auto-Encoder (AE)
Stacked Auto-Encoder (SAE)
Restricted Boltzmann Machines (RBM)
Deep Belief Network (DBN)
Classification experiment based on SAE model
Comparison of traditional methods and the AE
SAE model hidden layer neuron tuning
Building SAE based on AE
Classification experiments based on DBN model
Comparative analysis of the RBM and the traditional method
DBN model hidden layer neuron tuning
Building DBN based on RBM
Optimal model accuracy evaluation
Optimization Strategy

Figure 1: Technical flow.

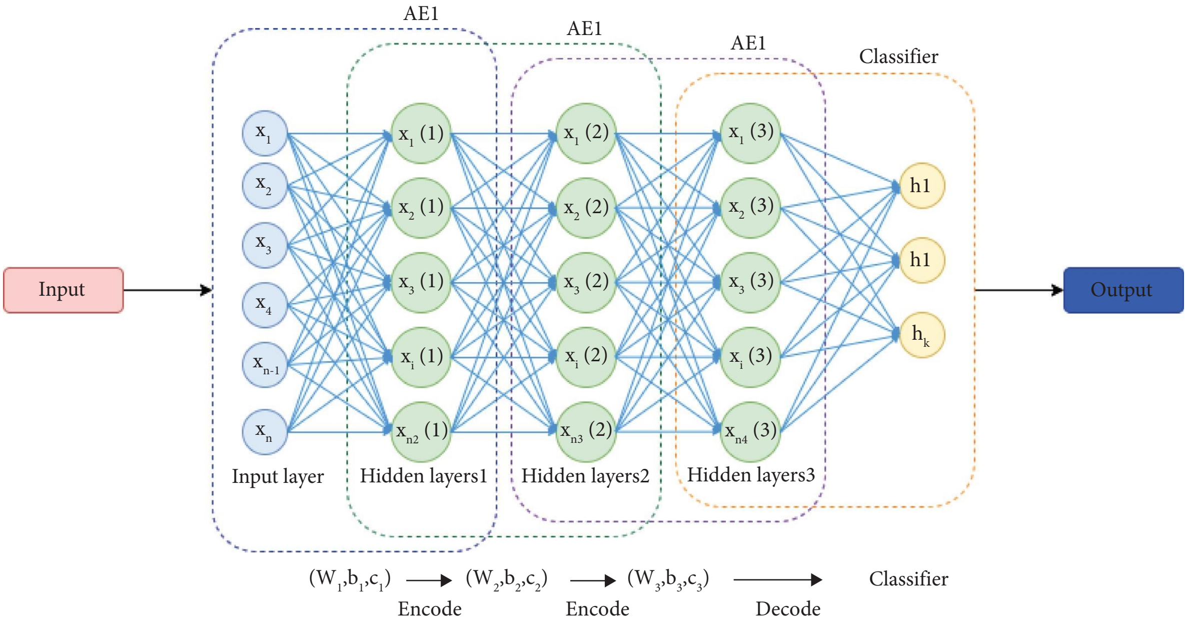


Figure 2: Schematic diagram of the stacked autoencoder (SAE) model.

number of neurons in the input layer) in the input layer to generate $x \in R^{n2}$ ($n2$ denotes the number of neurons in the hidden layer) through a linear function containing trainable parameters $W_1$ and $B_1$ and an activation function $f(x)$ (equation (1)) (generic activation functions are sigmoid functions, as in equation (4)). The encoder performs this step in the neural network called encoding. $X$ is then mapped to the output layer by an activation function $g(x)$ and a linear function containing trainable parameters $W_2$ and $B_2$ to produce $Z$ (equation (2)), so that the input $x$ approximates $Z$.

This step is performed by the decoder, which is called reconstruction. Sometimes, a linear-decoding approach can also be used to remove the activation function $g(x)$, i.e., equation (3):

$$X = f(W_1 x + B_1), \tag{1}$$

$$Z = g(W_2 X + B_2), \tag{2}$$

$$Z = W_2 X + B_2, \tag{3}$$

$W_1$ and $W_2$ denote implied input and implied output weights, respectively, and $B_1$ and $B_2$ denote offsets.

$$f(x) = \frac{1}{1 + \exp^{-x}}. \tag{4}$$

*2.2.2. Stacked Autoencoder (SAE).* The training process of a stacked autoencoder (SAE) is essentially unsupervised learning. The model is trained by layer-wise pretraining, which means that the parameters obtained in the first layer are propagated forward to obtain the first hidden layer. Then, it is used as the input layer to train the parameters in the second layer. Training is iterated gradually to guide the training of the parameters in each layer. The training parameters of each layer in the SAE are obtained. Finally, fine-tuning is performed by the connected classifier. Fine-tuning refers to treating all layers of the SAE as a single model and optimizing all weights in the network by an iterative algorithm using a label training sample set. A backpropagation algorithm with autocoding is applied to update the weights, which can be extended to apply as many layers as desired due to the application of the backpropagation algorithm (Algorithms 1 and 2).

### *2.2.3. Stacked Autoencoder (SAE) Algorithm Flow*

## *2.3. Deep Belief Network*

*2.3.1. Restricted Boltzmann Machines.* An RBM is a randomly generated neural network that learns the probability distribution of the original data. It is a typical energy-based model with the basic structure of a two-part diagram with one layer for the data input layer and one layer for the hidden layer; additionally, the nodes in each layer are not connected, but the nodes between the layers are fully connected. All the nodes can only take values of 0 and 1. They are random binary variable nodes, and this complete probability distribution $E(v, h)$ satisfies the Boltzmann distribution. This model is the RBM:

$$E(v, h) = -b^{'}v - c^{'}h - h^{'}Wv, \tag{5}$$

$W$ denotes the connection weight between the visible and hidden layers and $b$ and $c$ represent the respective biases of the visible and invisible layers.

*2.3.2. Deep Belief Network.* The DBN consists of multiple layers of the RBM superimposed to extract deep features of the original data (Figure 3). The joint probability distribution between the input data $v$ in the visual layer and the l-layer hidden layer $h^k$ is shown in equation (6). The weights are obtained using unsupervised greedy algorithm (GA) training, by first training the first layer of the RBM and fixing its training parameters, and then using the output of the hidden layer of the first layer of the limited Boltzmann machine as the input of the second layer of the RBM, in turn, training the parameters layer by layer. The last hidden layer is connected to the classifier. Refinement is completed by the supervised gradient descent (GD) algorithm, whose algorithm flow is as follows: except for the first layer RBM, the weights of the remaining RBM can be classified as upward cognitive weights as well as low generative weights. The generation process, after the top layer representation and assignment of low weights to form the bottom layer state, while refining the upward weights between layers, finally obtains the classification model based on DBNs:

$$P\left(v, h^1, h^2, \cdots, h^l\right) = \left(\sum_{k=0}^{l=2}\left(Ph^k \rangle h^{k+1}\right)\right)P\left(h^{l-1}, h^l\right), \tag{6}$$

where $P(h^{l-1}, h^l)$ is the joint probability distribution between the topmost RBM model visible layer and the hidden layer (Algorithms 3 and 4).

### *2.3.3. DBN Algorithm Flow*

*2.4. Support Vector Machines (SVMs).* SVMs are supervised learning algorithms commonly used for statistical and regression analysis. They are linear classifiers that find the maximum spacing on the feature space. Their goal is to maximize spacing, by which the optimization problem becomes a convex quadratic programming problem. In addition to being able to be used for linear classification, support vector machines can perform nonlinear classification, where vectors are mapped into a high-dimensional space that establishes a maximum spacing hyperplane. For example, in the separated hyperplane of the divided data, there are two parallel hyperplanes whose planes are parallel. The separated hyperplane maximizes the distance between these two hyperplanes if the distance between the hyperplanes is far, implying a small overall error.

In the given training sample set $(x_i, y_i)$, $y_i$ equals −1 or 1, which is the label of xi. Each $x_i$ is a p-dimensional real vector requiring a "maximum hyperplane" between the set of points with $y_i = -1$ and the set of issues with $y_i = 1$, so that the distance between the nearest point xi and the hyperplane is maximized. Any hyperplane satisfies equation (7), and the distance between them can be maximized if the training data are linearly separable, i.e., if the two hyperplanes can be used to separate the two types of data.

The support vector machine is extended to linearly indistinguishable using a loss function (equation (7)). The objective function is as in equation (8). The parameter $\lambda$ is used to weigh the relationship between increasing the size of the interval and ensuring that *xi* lies on the right side of the interval. Thus, for sufficiently small values of $\lambda$, it is assumed that the original data can be classified linearly even if it is not linearly classifiable, and still, a feasible classification criterion

```
Step 1: start
Step 2: given the set of training samples, the number of cells in the visible and hidden layers, the number of iterations, the learning rate, the initialized training parameter weight matrix $W$, the bias vectors $b$, $c$, and the canonical terms
Step 3: using the restricted Newton's method algorithm, update the training parameters until the algorithm converges
Step 4: closing
```

Algorithm 1: Autoencoder (AE) training algorithm.

```
Step 1: start
Step 2: given the parameters in Algorithm 1 and the number of hidden layer layers
Step 3: train the first layer of AE; the training algorithm is as in Algorithm 1
Step 4: use the hidden layer of the first layer of AE as the input layer of the second layer of AE and train layer by layer in turn until the last layer of AE
Step 5: connect the last output layer to the classifier to complete fine-tuning
Step 6: closing
```

Algorithm 2: Stacked autoencoder (SAE) training algorithm.

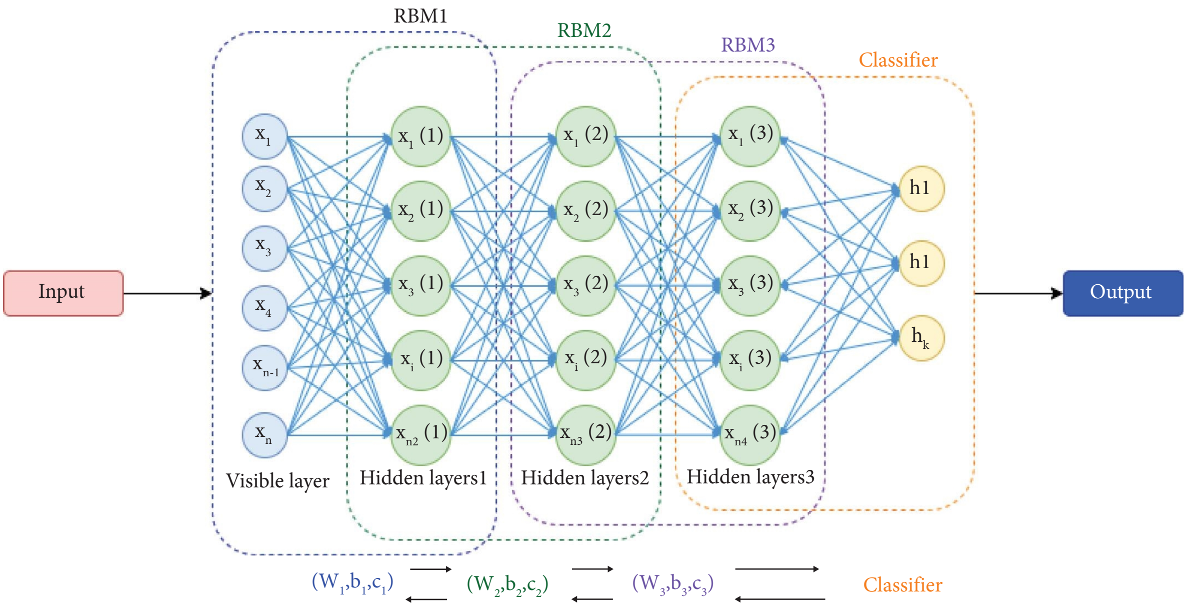


Figure 3: Schematic diagram of the deep belief network (DBN) model.

```
Step 1: start
Step 2: given the set of training samples, the number of cells in the visible and hidden layers, the number of iterations, the learning rate, the initialized training parameter weight matrix $W$, and the bias vectors $b$ and $c$
Step 3: using the contrast scattering algorithm, update the training parameters until the algorithm converges
Step 4: closing
```

Algorithm 3: RBM training algorithm.

```
Step 1: start
Step 2: given the parameters in Algorithm 1 and the number of hidden layer layers
Step 3: train the first layer of RBM and the training algorithm as in Algorithm 1
Step 4: use the hidden layer of the first RBM layer as the input layer of the second RBM layer and train layer by layer in turn until the
last RBM layer
Step 5: connect the final layer of output to the classifier to complete fine-tuning
Step 6: closing
```

Algorithm 4: DBN training algorithm

can be learned. Support vector machines are generalized linear classifiers, and as such, the choice of kernel functions and kernel parameters significantly impacts their performance:

$$w \cdot x - b = 0, \tag{7}$$

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(w \cdot x - b)\right)\right] + \lambda\|w\|^2. \tag{8}$$

*2.5. Softmax Regression Classifier.* The softmax regression classifier is an extension of the logistic regression classifier to this problem of multiple classifications. The test input $x$ takes a hypothesis function to estimate the probability value for each category $j$.

The objective function is as in equation (9), where l{ } is the demonstrative function with the rule that l{ } = 1, expression is true, and l{ } = 0, expression is false. The probability that $x$ is classified as $j$ in softmax regression is given in equation (10). The objective function is optimized using the gradient descent method. Equation (11) can be substituted into an algorithm such as the gradient descent method to minimize the objective function (10):

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\}\log\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k}\theta_l^T x^{(i)}}\right], \tag{9}$$

$$p\left(y^{(i)} = j \,\middle|\, x^{(i)};\theta\right) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}, \tag{10}$$

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\left\{y^{(i)} = j\right\} - p\left(y^{(i)} = j \,\middle|\, x^{(i)};\theta\right)\right)\right]. \tag{11}$$

*2.6. Sparse Autoencoder.* The sparse representation can be defined as follows: when the value of the output neuron is close to 1, it is defined as neuron activation, and when the value of the output neuron is close to 0, it is defined as neuron inhibition. When a large number of neurons in the hidden layer are inhibited, and only a small number of neurons are stimulated, this is the sparse state; i.e., a large number of components in the feature vector are 0. The mathematical expression is equation (12), which indicates that the $j^{th}$ component of the Nth data has as few nonzero terms as possible (e.g., $\rho = 0.05$). The penalty factor of sparse representation is added to the original objective function, and the loss function of sparse representation is equation (13), where $n2$ denotes the number of neurons in the hidden layer and $j$ denotes each neuron in the hidden layer. Actual relative entropy KL divergence is also expressed as the relative entropy between two Bernoulli random variables, as shown in equation (14). When the two variables are equal, the relative entropy equals 0. When the difference between the two variables becomes larger, the relative entropy increases until it approaches infinity. Therefore, minimizing the penalty factor can also make $\rho$ and $\rho_i$ close. The loss function of the original optimization problem can be expressed in equation (15), and $\mu$ denotes the weight of the penalty factor of the sparse term:

$$\rho_i \overset{\Delta}{=} \frac{1}{N}\sum_{k=1}^{N} X_j(k) = \rho, \tag{12}$$

$$\sum_{j=1}^{n_2} KL\left(\rho \,\|\, \rho_i\right), \tag{13}$$

$$KL\left(\rho \,\|\, \rho_i\right) = \rho\log\frac{\rho}{\rho_i} + (1-\rho)\log\frac{1-\rho}{1-\rho_i}, \tag{14}$$

$$\begin{aligned} J\left(W_1, B_1, W_2, B_2\right) = {} & \frac{1}{N}\sum_{k=1}^{N}\|Z(k) - x(k)\|^2 \\ & + \lambda\left(\|W_1\|_F^2 + \|W_2\|_F^2\right) + \mu\sum_{j=1}^{n_2}\mathrm{KL}\left(\rho \,\|\, \rho_i\right). \end{aligned} \tag{15}$$

## 3. Data

*3.1. Data Source.* This study used two sets of hyperspectral image data simultaneously for the experiments. The purpose of selecting two sets of different data is to verify the model and method's reliability and the experimental results' generalizability and extensiveness. The two sets of data selected have different feature types and spatial resolutions. The two datasets are airborne aerial hyperspectral image data from Pavia, Italy, and ground-based close-range hyperspectral image data acquired using the HySpex imaging spectrometer.

The first dataset is the publicly available hyperspectral data from the city of Pavia, Italy, acquired by the airborne reflective optic system imaging spectrometer (ROSIS-3). This image is 610 × 340 pixels (Figure 4(a)). The ROSIS-3 sensor generates a spectral range of 430–860 nm and 115 bands. The main categories are asphalt, bare ground, gravel, grass, metal sheets, brick, trees, and shadows.

The second dataset is the ground-based near-field hyperspectral image data from the Chengdu University of Technology acquired by using the HySpex imaging spectrometer. The image is 400 × 600 pixels (Figure 4(b)). The HySpex sensor generates 1600 spatial pixels, with a spectral range of 400–1000 nm, and 108 bands. The main features are water, vegetation, concrete roads, lava rocks, steel, glass, and walls.

*3.2. Data Preprocessing.* The ROSIS-3 data are preprocessed that can be used immediately in experiments. The HySpex data are radiometrically corrected by the radiometric calibration module that comes with the HySpex imaging spectrometer, and reflectance inversion was performed by the flat-field method based on the statistical model; a large concrete field was used as a flat field. The image data can be used for experiments after preprocessing.

## 4. Results

*4.1. Sample Selection.* We used the two types of image data to select a total of eight feature classes. The selection of each type of ground object sample data in the ROSIS-3 data is shown in Table 1, and the spectral curve of each type of ground object sample is shown in Figure 4(a). The sample data of each type of ground object in the HySpex data are shown in Table 1, and the sample spectrum curve of each type of ground object is shown in Figure 4(b). The selected samples were divided into training, validation, and test samples in a ratio of roughly 3 : 1 : 1. We used the training samples to adjust the trainable parameters of the model, the validation sample to adjust the hyperparameters of the model, and the test sample to test the classification accuracy of the model Figure 5.

### *4.2. Classification Experiment Based on the Stacked Autoencoder (SAE)*

*4.2.1. Analysis of the Autoencoder (AE) Model.* AEs with different numbers of neurons in the hidden layer were trained separately with the same training samples. AE training is aimed at making the original and the reconstructed data as similar as possible after encoding and decoding. Thus, the model's performance can be analyzed by the reconstructive ability of the model on the test samples after the training samples are completed. The AE coded and decoded the first 100 features of the test sample to obtain the reconstructed features. The original and reconstructed features were transformed into a $10 * 10$ pixel-size frame to represent the image features. We selected representative asphalt and grassland features in the ROSIS-3 data for the experiment. The visualization experiment results of the features reconstructed using different numbers of neurons in the hidden layer are shown in Figures 6(a) and 6(b). We selected representative water bodies and concrete road features in the HySpex data for the second experiment. The visualization experiment results of the features reconstructed with different numbers of hidden layer neurons are shown in Figures 6(c) and 6(d). The spectral curve of asphalt is obtained, whose reflectance has been stabilized at a lower level, so the image is mainly blue, and the analysis can be performed when the number of neurons in the hidden layer is 30. The best effect of reconstruction is obtained when the number of neurons in the hidden layer is 30; the spectral curve of grass is obtained, whose reflectance has been kept at a lower state, when the wavelength is below 700 nm, and when the wavelength is 700 nm, an obvious jump is produced. Reflectance suddenly increases, and then, the region is stabilized. The image is blue first and then transitions quickly to red. The reconstruction effect is best when the number of neurons in the hidden layer is 50 and the number of neurons in the hidden layer is 30. In summary, the best reconstruction effect is achieved when the number of neurons in the hidden layer of the AE is 30–50, and AE performance is the best at this time.

*4.2.2. Comparison of Traditional Methods with AEs.* After the analysis has yielded a better selection of the number of neurons in the hidden layer of the AE, it is natural to verify whether it can help the classifier's classification accuracy. The AE is compared and analyzed with several commonly used downscaling and feature extraction algorithms. The classifiers connected after the feature extraction algorithm are an SVM classifier and the softmax regression classifier. The SVM kernel function is chosen as a radial basis kernel function suitable for hyperspectral image classification. The hyperparameter $\sigma$ of the kernel function is set to 0.009; the penalty parameter is 100. The learning rate of softmax regression is chosen to be 0.1, and the optimization iteration number is 500. The number of neurons in the first hidden layer is set to 40, and the number of neurons in the second layer is [5, 10, 15, 20, 25, and 30]. We selected the traditional downscaling and feature extraction algorithms as principal component analysis, minimum noise fraction rotation (MNF Rotation), factor analysis (FA), and independent components. The experimental results of comparing the classification accuracy between AEs and traditional methods with ROSIS-3 data are shown in Figures 7(a) and 7(b). The analysis shows that the classification accuracy of AEs is the highest when the SVM is connected. The accuracy changes little with a change in the number of features. The classification accuracy of the factor analysis and principal component analysis is similar and stable but still significantly lower than that of the AE. When the softmax regression classifier is connected, the classification accuracy of the AE is further improved, and the classification accuracy reaches the optimum value (>90%) when the number of features is 20. The experimental results of comparing the classification accuracy between AEs and

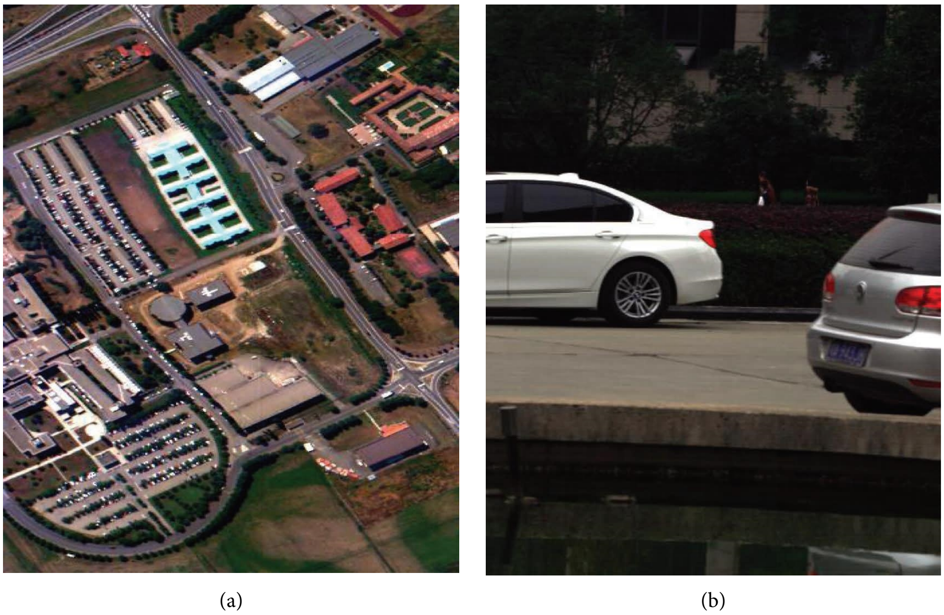


Figure 4: Hyperspectral images used in the study: (a) data from a true-color composite ROSIS-3 image; (b) HySpex composite image data in true color.

Table 1: Selection of sample data for ROSIS-3 and HySpex data (number).

| | Training samples | Validation samples | Test samples | Total |
|---|---|---|---|---|
| Asphalt (*R*) | 1345 | 450 | 502 | 2297 |
| Bare ground | 831 | 282 | 335 | 1448 |
| Gravel | 1193 | 400 | 458 | 2051 |
| Grassland | 1002 | 333 | 385 | 1720 |
| Metal sheet | 1850 | 625 | 762 | 3237 |
| Brick | 952 | 333 | 389 | 1674 |
| Shadows | 2247 | 718 | 759 | 3724 |
| Trees | 1380 | 459 | 510 | 2349 |
| Water (*H*) | 1148 | 379 | 390 | 1917 |
| Vegetation | 1161 | 388 | 383 | 1932 |
| Concrete road | 1321 | 446 | 449 | 2216 |
| Magma rock | 1007 | 351 | 356 | 1714 |
| Steel | 1085 | 373 | 377 | 1835 |
| Glass | 633 | 211 | 211 | 1055 |
| Wall | 1233 | 411 | 411 | 2055 |

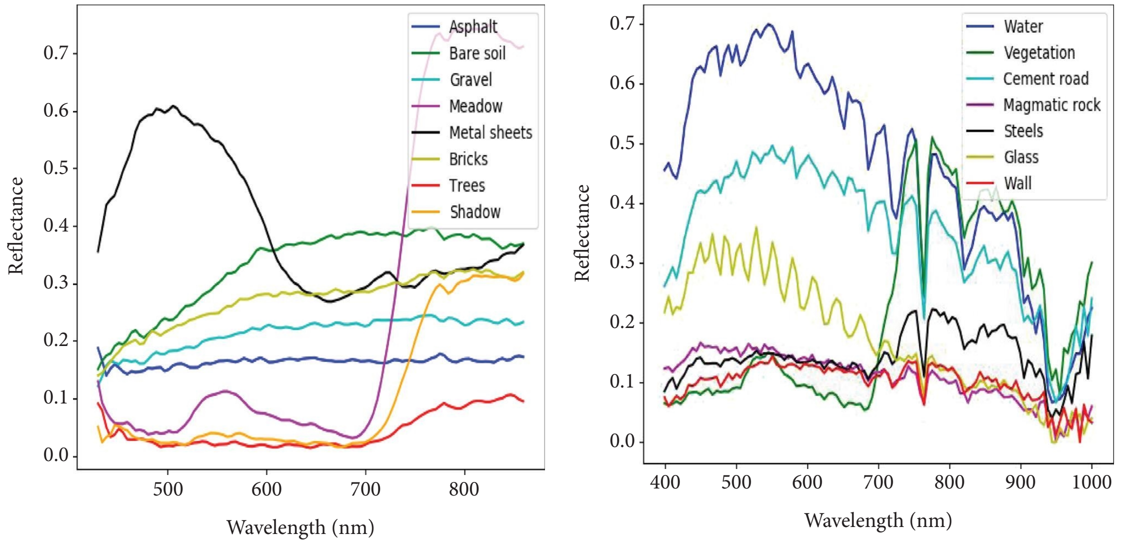
*Note. R* is for ROSIS-3 data; *H* is for HySpex data.

traditional methods with the HySpex data are shown in Figures 7(c) and 7(d). The analysis shows that the classification accuracy of the AE is the highest when the SVM classifier is connected. The classification accuracy increases slowly as the number of features increases. When the number of features is too large, the classification accuracy decreases significantly.

In contrast, the classification accuracy of the factor analysis and principal component analysis remains stable. Still, the classification accuracy is lower than that of AEs, and independent component analysis and minimum noise separation accuracy are still more inadequate. However, when the softmax regression classifier is connected, the classification accuracy of the AE is improved and the classification accuracy shows a steady increase. Its classification accuracy is better than other methods at this time.

According to the analysis above, the AE performs significantly better overall than the other four feature extraction algorithms. The classification accuracy exhibits a more stable trend as the number of features changes. Additionally, the classification accuracy is higher when the softmax regression classifier is connected than when the support vector machine classifier is connected.

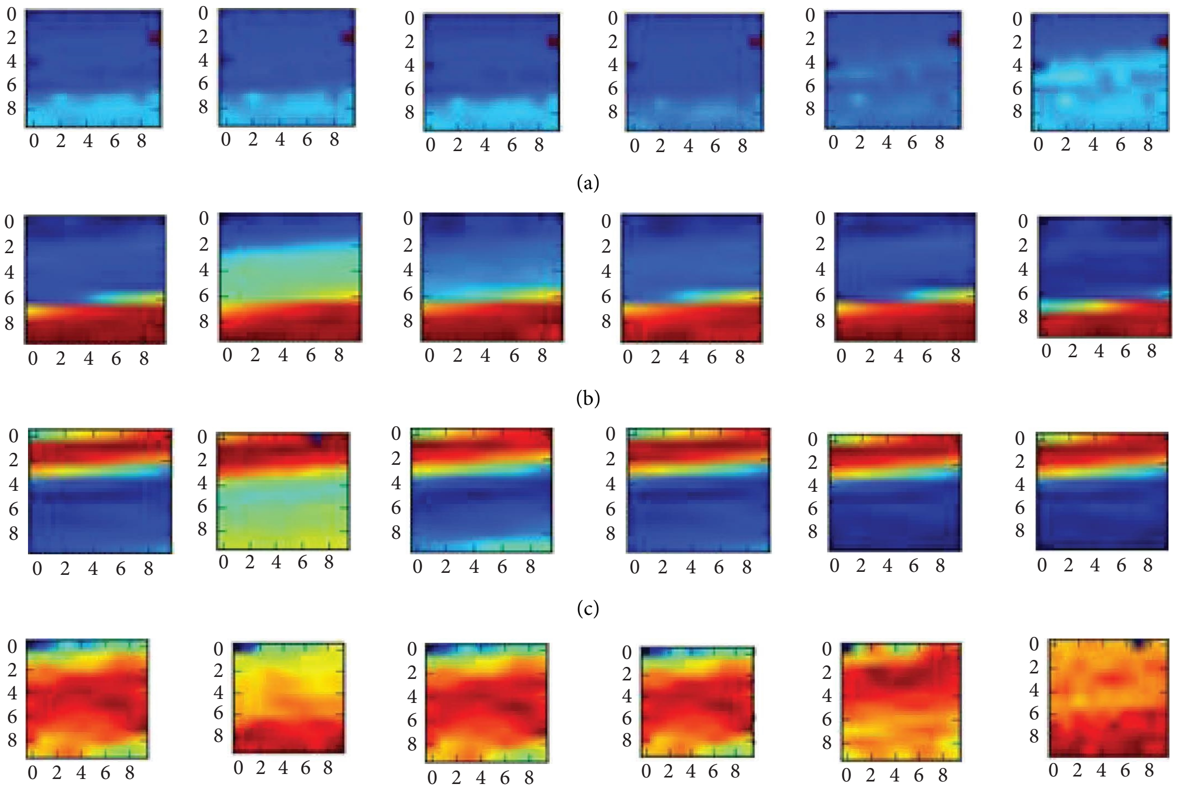
*4.2.3. Analysis of the Impact of the Number of Hidden Layers in SAEs.* The conclusion of the previous section demonstrated that the feature extraction ability of the AE is better than that of the traditional feature extraction algorithm and



(a)

(b)

Figure 5: The sample data spectrum character curves: (a) the sample data spectrum character curve of ROSIS-3; (b) the sample data spectrum character curve of HySpex.



(a)

(b)

(c)

(d)

Figure 6: Reconstructed features under ROSIS-3 data with various numbers of hidden layer neurons. (a) Original and recreated asphalt spectral curve visualizations, when the concealed layer has 10, 30, 50, 70, or 90 neurons, respectively, are shown from left to right. (b) Original and reconstructed versions of the grass spectral curve, left to right, with 10, 30, 50, 70, and 90 neurons in the hidden layer. Visualization of reconstructed features with the different number of hidden layer neurons under HySpex data. (c) Original and reconstructed water body spectral curve with 10, 30, 50, 70, and 90 neurons in the hidden layer, from left to right. (d) Visualization of the original and reconstructed cement road spectral curve with 10, 30, 50, 70, and 90 neurons in the hidden layer, from left to right. The color transitions from blue (low reflectance) to red (high reflectance).

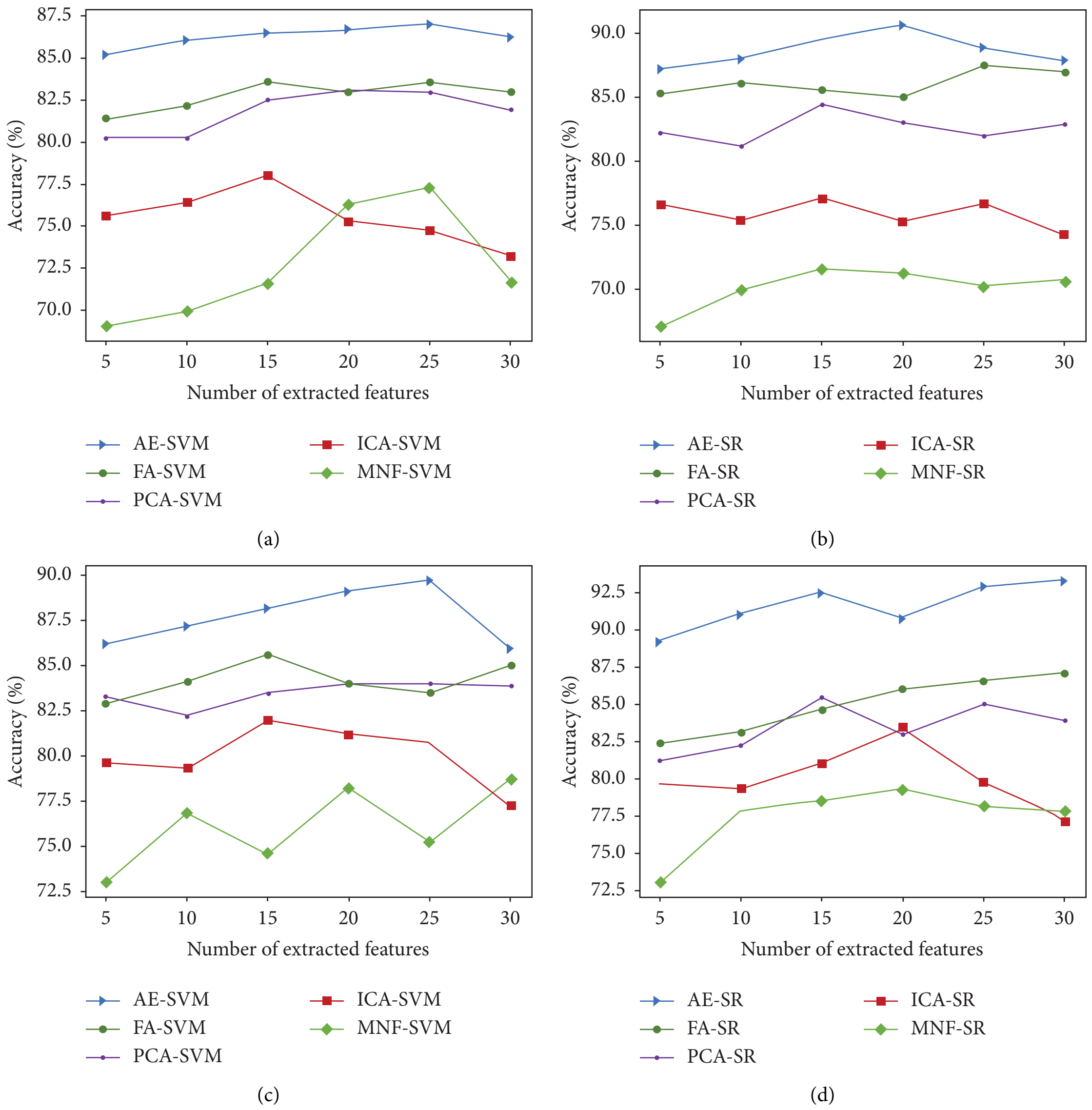


FIGURE 7: (a, b) Comparison of classification accuracy between autoencoders (AEs) and traditional methods under ROSIS-3 data. (c, d) Comparison of classification accuracy between autoencoders (AEs) and traditional methods under HySpex.

that the classification accuracy is better when the AE is connected to the softmax regression classifier. Next, the SAE is constructed by the AE, and the classification model is built by combining the softmax classifier and comparing the effect of different layers of the AE on the classification accuracy.

One of the critical factors determining the performance of SAEs is the number of hidden layers, that is, the selection of the number of AE layers. The number of different AE layers determines what kind of features is extracted and plays a critical role in the final classification accuracy. When the number of layers is too small, only shallow parts can be removed, which affects the image classification accuracy, and as the number of hidden layers increases, more and more abstract feature representations can be obtained. However, when the number of layers is too large, the model may be overfitted [67]. This shows the impact of the number of layers of the AE on the classification performance. From the above section, it can be tentatively determined that better classification accuracy can be obtained when the number of neurons in the hidden layer is 30–50. Thus, the number of hidden layer neurons is fixed at 40, and the number of AE layers is selected as [1–5], respectively, to confirm the reliability of the experiment by repeating the experiment several times under a single condition, which is finally represented by a box plot. The classification accuracy results of the SAE with different hidden layers of the ROSIS-3 data are shown in Figure 8(a). Figure 8(b) shows the classification accuracy results of the stacked autoencoder for the HySpex data with different numbers of hidden layers. As the number of layers of the AE increases from 1 to 3, the classification accuracy increases significantly and the stability of the classification accuracy increases gradually, and with further increases in the number of layers of the AE, the classification accuracy starts to show a specific decreasing trend. When the

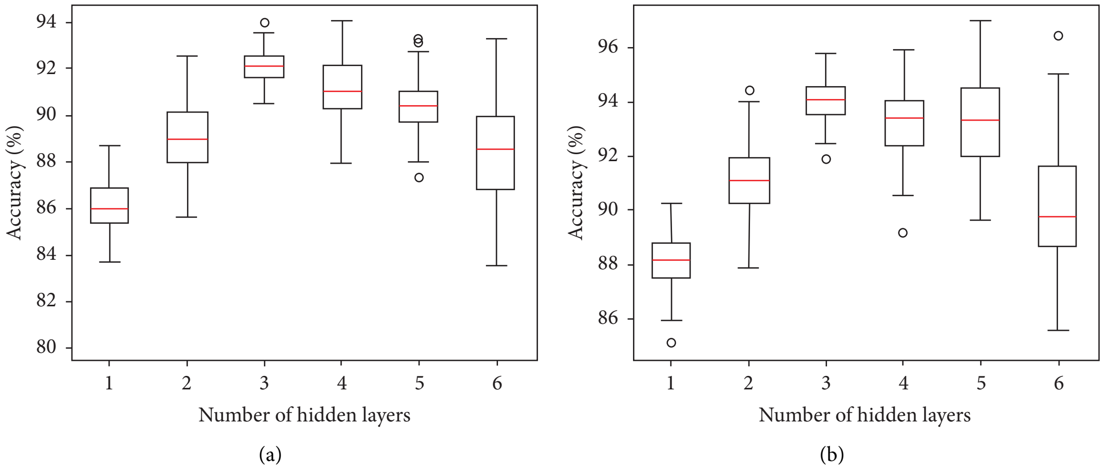


Figure 8: (a) Box plot of classification accuracy of the stacked autoencoder (SAE) with the different number of hidden layers under ROSIS-3 data. (b) Box plot of the classification accuracy of the SAE with the different numbers of hidden layers under HySpex data.

number of layers of the AE reaches 6, the classification accuracy decreases significantly and the classification accuracy starts to become unstable. The median classification accuracy is 92.12% for the ROSIS-3 data and 94.02% for the HySpex data.

*4.2.4. Optimal Model Accuracy Evaluation.* According to the experimental analysis, the optimal classification model is obtained when the number of neurons in the hidden layer is 40 and the number of layers in the AE is 3. Based on the conclusion of the experiments in the previous section, two experiments with a classification accuracy of 92.12% under the ROSIS-3 data and 94.02% under the HySpex data were selected to evaluate the accuracy. The confusion matrix of the SAE-SR classification accuracy with the ROSIS-3 data is shown in Table 2. The analysis shows that the classification model is better at recognizing metal sheets, grass, and trees, performs worse for asphalt, gravel, and bare ground, and has an average performance for bricks and shadows, which are easily confused with asphalt, bare ground, and gravel, and between shadows and trees. The confusion matrix of the SAE-SR classification accuracy of the HySpex data is shown in Table 3. Except for steel plates and concrete roads, the recognition degree of the other features is higher; steel plates and walls are easily confused.

### *4.3. Classification Experiment of Classification Model Based on Deep Belief Networks*

*4.3.1. Analysis of the RBM Model.* Using two types of image data, ROSIS-3 and HySpex, we applied an RBM model of the structural unit of the deep belief network. We analyzed the influence of the number of hidden layer neurons on the performance of the RBM, where the hyperparameters in the RBM were designed, with a learning rate of 0.1, according to the hyperparameter selection advice given in [68]. The number of hidden layer neurons was set as [10, 30, 50, 70, 90], respectively. The RBM containing different numbers of neurons in the hidden layer was trained with the same training samples separately until the algorithm converged. In this section, the model's performance is determined by the ability of the model to reconstruct the test samples after training was completed.

We compared the spectral curves of the original sample with the reconstructed spectral curves under different experimental parameters. The difference in reconstructing ability of the RBM with different numbers of hidden layer neurons can be visually compared in Figure 9. Representative asphalt and grassland features in the ROSIS-3 data were selected for the experiment. The spectral curve results of the experiment are shown in Figure 9. When the number of hidden layer neurons is 30, the reconstructed asphalt spectral curve is most similar to the original asphalt spectral curve and the reconstructed grass spectral curve is most similar to the original grass spectral curve. The reconstructed cement road spectral curve is most similar to the original cement road spectral curve. In summary, the performance of the RBM is optimal when the number of hidden layer neurons is 30. A too large or too small number of hidden layer neurons have a large impact on the performance.

*4.3.2. Comparison of the RBM with the Traditional Method.* The impact of the number of neurons in the hidden layer on the performance of the RBM was analyzed. The best performance of the RBM was found when the number of neurons in the hidden layer was 30. The best performing RBM was compared with the AE and the traditional feature extraction algorithm. According to the conclusion obtained in the experiment in Section 4.2.3, the classification accuracy of the two traditional feature extraction algorithms (factor analysis and principal component analysis) was higher, so only these two algorithms are selected as the traditional feature extraction algorithm in this section of the experiment. In the experiments, two neural network layers of the RBM were constructed; the number of neurons in the first hidden layer was set to 30, and the numbers of neurons in the

Table 2: Confusion matrix for the SAE-SR classification accuracy of ROSIS-3 data.

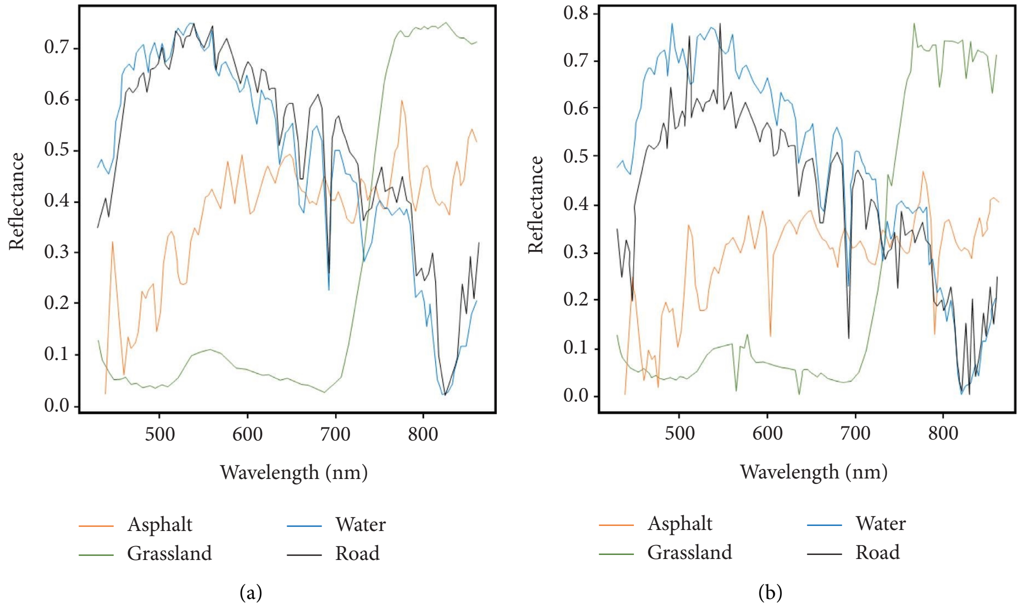
| | Asphalt | Bare ground | Gravel | Grassland | Metal sheet | Brick | Shadows | Trees | Total | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Asphalt | 422 | 10 | 49 | 0 | 0 | 21 | 0 | 0 | 502 | 84.06 |
| Bare ground | 0 | 295 | 40 | 0 | 0 | 0 | 0 | 0 | 335 | 88.06 |
| Gravel | 19 | 30 | 401 | 0 | 0 | 8 | 0 | 0 | 458 | 87.55 |
| Grassland | 0 | 0 | 0 | 371 | 0 | 0 | 12 | 2 | 385 | 96.36 |
| Metal sheet | 0 | 0 | 4 | 0 | 741 | 17 | 0 | 0 | 762 | 97.24 |
| Brick | 16 | 14 | 1 | 0 | 0 | 358 | 0 | 0 | 389 | 92.03 |
| Shadows | 13 | 0 | 0 | 0 | 0 | 0 | 701 | 45 | 759 | 92.36 |
| Trees | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 488 | 510 | 95.69 |
| Total | 470 | 349 | 495 | 371 | 741 | 404 | 735 | 535 | | |
| Accuracy (%) | 89.79 | 84.52 | 81.01 | 100 | 100 | 88.61 | 95.37 | 91.21 | | |

Overall accuracy = 3777/4100 = 92.12%; kappa coefficient = (0.9212 – 0.1353)/(1 – 0.1353) = 0.91.

Table 3: Confusion matrix for the SAE-SR classification accuracy of HySpex data.

| | Water | Vegetation | Concrete road | Magma rock | Steel plate | Glass | Wall | Total | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| Water | 377 | 8 | 5 | 0 | 0 | 0 | 0 | 390 | 96.67 |
| Vegetation | 18 | 365 | 0 | 0 | 0 | 0 | 0 | 383 | 95.30 |
| Concrete road | 7 | 30 | 412 | 0 | 0 | 0 | 0 | 449 | 91.94 |
| Magma rock | 0 | 0 | 0 | 335 | 3 | 0 | 18 | 356 | 94.10 |
| Steel plate | 0 | 0 | 0 | 0 | 335 | 8 | 34 | 377 | 88.86 |
| Glass | 0 | 0 | 0 | 0 | 0 | 201 | 10 | 211 | 95.26 |
| Wall | 0 | 0 | 0 | 3 | 5 | 5 | 398 | 411 | 96.84 |
| Total | 402 | 403 | 417 | 338 | 343 | 214 | 460 | | |
| Accuracy (%) | 93.78 | 90.57 | 98.80 | 99.11 | 88.86 | 93.93 | 86.52 | | |

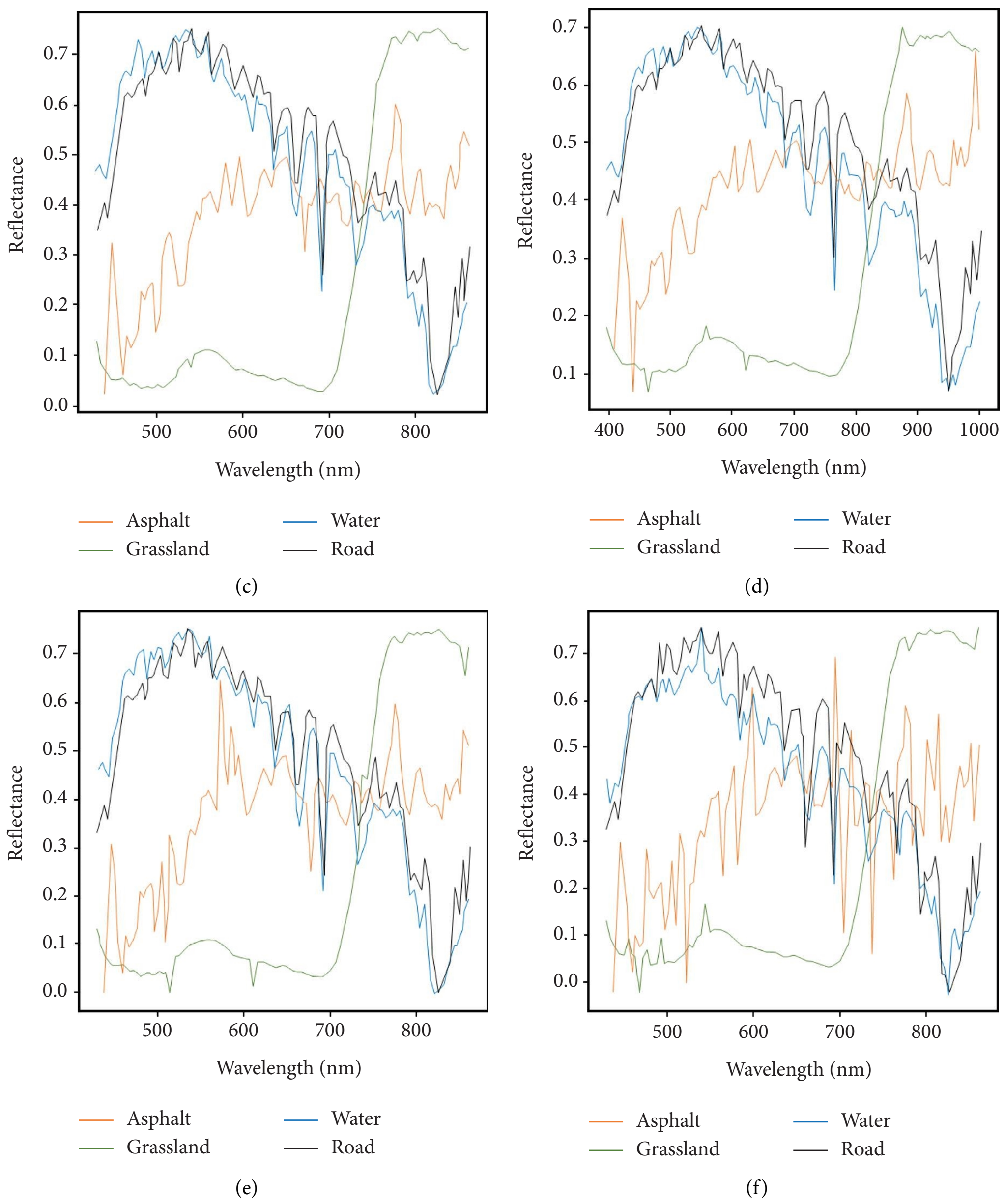
Overall accuracy = 2423/2577 = 94.02%; kappa coefficient = (0.9402 – 0.1479)/(1 – 0.1479) = 0.93.



(a)

(b)

Figure 9: Continued.



(c)

(d)

(e)

(f)

Figure 9: Original and reconstructed spectral curves. (a) Original. (b) The number of neurons in the hidden layer is 10. (c) The number of neurons in the hidden layer is 30. (d) The number of neurons in the hidden layer is 50. (e) The number of neurons in the hidden layer is 70. (f) The number of neurons in the hidden layer is 90.

second layer were [5, 10, 15, 20, 25, and 30]. Then, two neural network layers of AEs were constructed; the number of neurons in the first hidden layer was set to 40, and the numbers of neurons in the second layer were [5, 10, 15, 20, 25, and 30]. The traditional dimensionality reduction methods (principal component analysis and factor analysis) extracted [5, 10, 15, 20, 25, and 30] features. The number of extracted features was set consistently for all dimensionality reduction methods and was connected to two classifiers: support vector machine and softmax regression classifier. The kernel function of the support vector machine was selected as a radial basis kernel function suitable for hyperspectral image classification; the hyperparameter $\sigma$ of the kernel function was set to 0.009, the penalty parameter was 100, the learning rate of the softmax regression was selected as 0.1, and the number of optimization iterations was 500. The experimental results for comparison of the classification accuracy of the RBM with other methods for the HySpex data are shown in Figure 10. The analysis shows that the classification accuracy of the RBM is significantly higher than that of the factor analysis and principal component analysis and is more stable with a change in the number of features; however, the classification accuracy of the RBM is overall inferior to that of the AE.

*4.3.3. Analysis of the Impact of the Number of Hidden Layers in DBNs.* From the conclusion of the previous section, it can

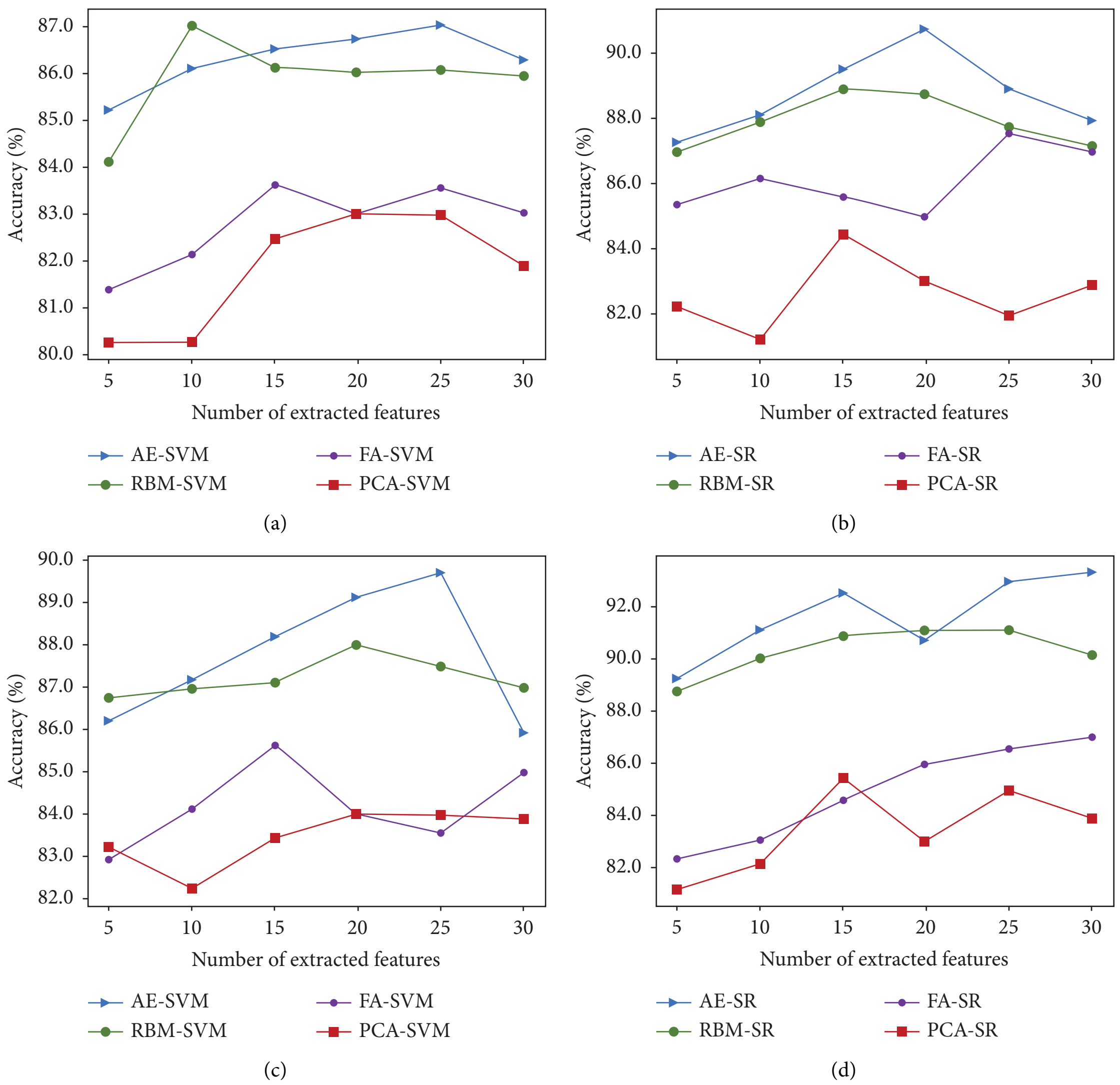
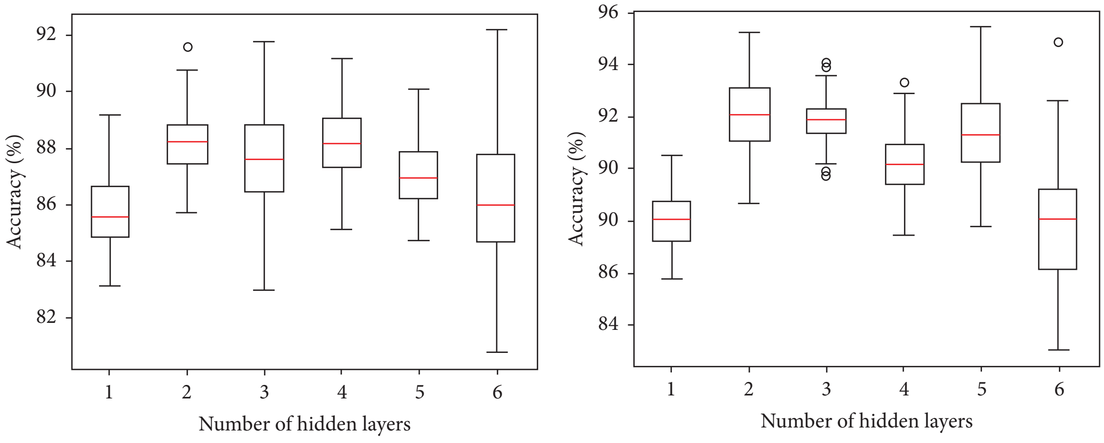


Figure 10: (a, b) Comparison of classification accuracy of restricted Boltzmann machines (RBMs) with other methods under ROSIS-3 data. (c, d) Comparison of classification accuracy of restricted Boltzmann machines (RBMs) with other methods under HySpex data.

be concluded that the feature extraction ability of the RBM is better than that of the two feature extraction methods of factor analysis and principal component analysis, and its feature extraction ability is inferior to that of the AE. The next step is to build a deep belief network by the RBM and connect the softmax classifier to construct the classification model.

The number of hidden layers of the deep belief network determines whether the appropriate features can be extracted, significantly impacting the final classification accuracy. When the number of layers is too small, only the shallow features can be extracted, improving the classification accuracy. However, when the number of layers is too large, it can produce overfitting. From the above section, it can be preliminary judged that the classification accuracy can be improved when the number of neurons in the hidden layer is 30. We fix the number of hidden layer neurons to 30 and select the number of hidden layers as [1–5], respectively; the reliability of the experiment is confirmed by repeating the experiment several times under a single condition, which is finally represented by the box plot. The experimental results of the classification accuracy of the ROSIS-3 data in different hidden layers of the deep belief network are shown in Figure 11(a). The classification accuracy of the HySpex data is shown in Figure 11(b). The median classification accuracy of the ROSIS-3 data is 88.22%, and the median classification accuracy of the HySpex data is 92.16%.

*4.3.4. Optimal Model Accuracy Evaluation.* According to the experimental analysis of the number of neurons in the hidden layer and the number of layers in the RBM, the optimal classification model is obtained when the number of neurons in the hidden layer is 30 and the number of layers in the RBM is 2. According to the conclusion of the experiments in the previous section, two experiments with a classification accuracy of 88.22% for the ROSIS-3 data



(a)

(b)

Figure 11: (a) Box plot of the classification accuracy for different numbers of hidden layers of the deep belief network under ROSIS-3 data. (b) Box plot of the classification accuracy for different numbers of hidden layers of deep belief networks under HySpex data.

and 92.16% for the HySpex data are selected to perform the confusion matrix of the ROSIS-3 data of DBN-SR classification accuracy shown in Table 4. The analysis shows that, compared with the SAE, there is a significant decrease in recognition of bare ground and bricks by the deep belief network and that there is also a certain decrease in the overall classification accuracy. The confusion matrix of the HySpex data of DBN-SR classification accuracy is shown in Table 5. In the confusion matrix of the DBN-SR classification accuracy, the analysis shows significant degradation in recognizing lava rocks compared to the SAE.

*4.4. Comparison of Two Optimal Models.* According to the experimental results obtained in the previous two sections, the optimal model of the SAE has a hidden layer with 40 neurons and three layers. The optimal model of the DBN has a hidden layer with 30 neurons and two layers. The comparative analysis of the box plots and the confusion matrix shows that SAE has higher classification accuracy and is more accurate and stable in the recognition of each feature. It can be concluded that the classification model based on the SAE and the softmax regression classifier is the optimal classification model obtained in this experiment. The optimal models of both algorithms were used to classify the entire image, and the classification results of the ROSIS-3 and HySpex data are shown in Figure 12. The analysis shows that, under the ROSIS-3 data, the SAE-SR classification model misclassifies some shadows and trees and some bare ground, grass, and trees. The DBN-SR classification model misclassifies bricks and gravels and confuses with shadows, trees, asphalt, and bricks, with a lower recognition accuracy of gravels and bricks compared with the SAE-SR classification model. Under the HySpex data, the SAE-SR classification model confuses vegetation, water, glass, and steel plates with wall. The DBN-SR classification model has a similar situation, but misclassification is more severe than in the SAE-SR classification model.

*4.5. Optimization.* According to the above box plot and accuracy evaluation analysis, we devised an optimization strategy using a stacked sparse autoencoder based on the SAE. The number of hidden layer neurons and hidden layers remained unchanged from the optimal parameters obtained which were described above, and the accuracy was compared with that of the ordinary SAE. The comparative analysis of the accuracy of the AE is shown in Table 6, which shows that the classification accuracy is improved by using the stacked sparse autoencoder.

The deep learning algorithm is computationally complex and takes a long time to train because of the many parameters in the deep learning model, which is a deep neural network, and many floating point and matrix operations are performed during the parameter-tuning phase of the model training process. The model computation process in this paper used GPU parallel computing. Figure 13 displays the analysis of the classification speed under GPU parallel computing outcomes. Basic computer hardware specifications in the experiments included an Intel i7-7500U processor, a GeForce GTX 960M graphic card, and 8 GB RAM. The software environment also included Ubuntu, cuda8.0, and cuDNN version 5. The model's training times for the two sets of image data (ROSIS-3 and HySpex) were reduced from 171 seconds to 23 seconds and from 321 seconds to 45 seconds, respectively. This significantly sped up image classification.

Table 4: Confusion matrix of DBN-SR classification accuracy for ROSIS-3 data.

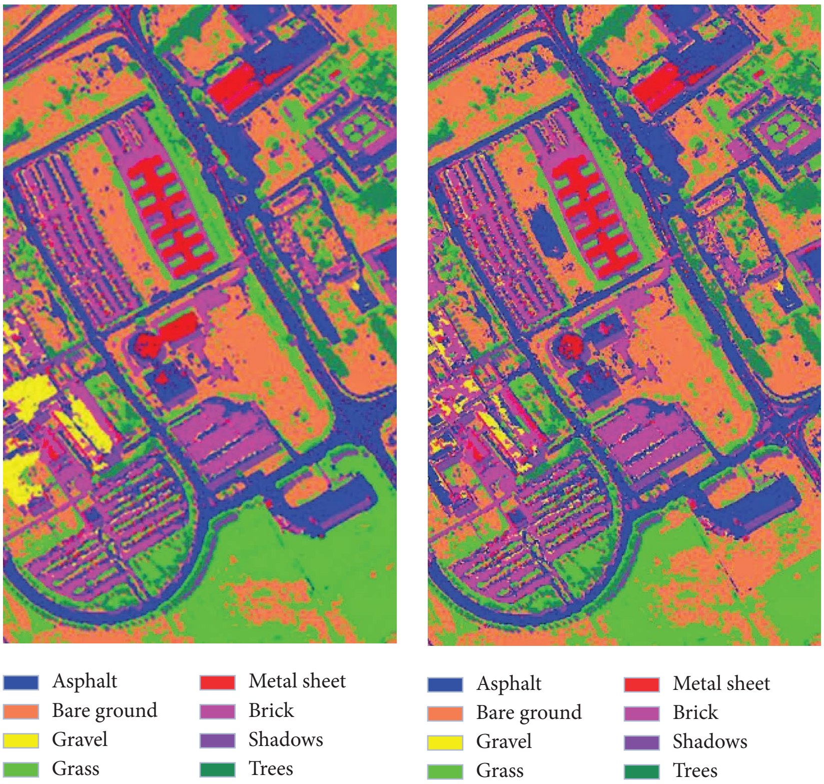
| | Asphalt | Bare ground | Gravel | Grassland | Metal sheet | Brick | Shadows | Trees | Total | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Asphalt | 426 | 19 | 31 | 0 | 0 | 26 | 0 | 0 | 502 | 85.53 |
| Bare ground | 0 | 275 | 45 | 0 | 0 | 15 | 0 | 0 | 335 | 81.82 |
| Gravel | 45 | 31 | 375 | 0 | 0 | 35 | 0 | 0 | 458 | 84.05 |
| Grassland | 0 | 0 | 0 | 321 | 0 | 0 | 42 | 22 | 385 | 94.01 |
| Metal sheet | 0 | 0 | 0 | 0 | 710 | 52 | 0 | 0 | 762 | 96.31 |
| Brick | 21 | 26 | 9 | 0 | 0 | 333 | 0 | 0 | 389 | 86.98 |
| Shadows | 11 | 0 | 0 | 0 | 0 | 0 | 698 | 50 | 759 | 93.95 |
| Trees | 0 | 0 | 0 | 0 | 0 | 0 | 31 | 479 | 510 | 95.33 |
| Total | 503 | 351 | 460 | 321 | 710 | 461 | 771 | 551 | | |
| Accuracy (%) | 84.69 | 78.35 | 81.52 | 100 | 100 | 72.23 | 90.53 | 86.90 | | |

Overall accuracy = 3617/4100 = 88.22%; kappa coefficient = (0.8822 − 0.1363)/(1 − 0.1363) = 0.87.

Table 5: Confusion matrix of DBN-SR classification accuracy for HySpex data.

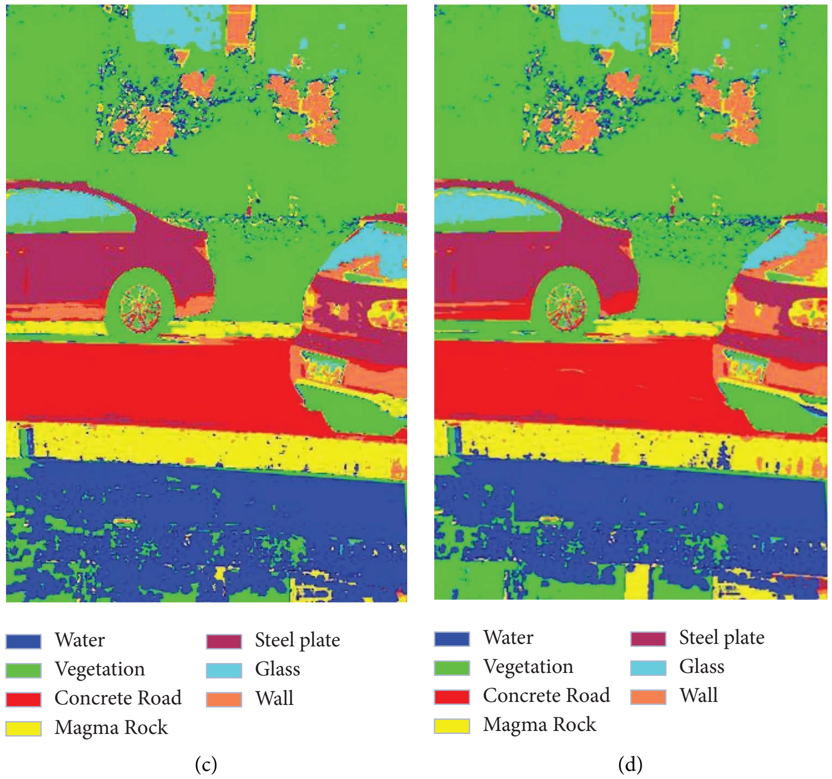
| | Water | Vegetation | Concrete road | Magma rock | Steel plate | Glass | Wall | Total | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| Water | 379 | 9 | 2 | 0 | 0 | 0 | 0 | 390 | 97.18 |
| Vegetation | 15 | 368 | 0 | 0 | 0 | 0 | 0 | 383 | 96.08 |
| Concrete road | 7 | 41 | 401 | 0 | 0 | 0 | 0 | 449 | 89.31 |
| Magma rock | 0 | 0 | 0 | 302 | 16 | 0 | 38 | 356 | 84.83 |
| Steel plate | 0 | 0 | 0 | 0 | 337 | 8 | 32 | 377 | 89.39 |
| Glass | 0 | 0 | 0 | 0 | 0 | 194 | 17 | 211 | 91.94 |
| Wall | 0 | 0 | 0 | 7 | 5 | 5 | 394 | 411 | 95.86 |
| Total | 401 | 418 | 403 | 309 | 358 | 207 | 481 | | |

Overall accuracy = 2375/2577 = 92.16%; kappa coefficient = (0.9216 − 0.1481)/(1 − 0.1481) = 0.91.



(a) (b)

Figure 12: Continued.



(c)

(d)

Figure 12: The data classification effect of ROSIS-3: (a) SAE-SR classification model; (b) DBN-SR classification model. The data classification effect of HySpex: (c) SAE-SR classification model; (d) DBN-SR classification model.

Table 6: Comparative analysis of the accuracy of the two autoencoders.

| Classification accuracy/(%) | ROSIS-3 data | HySpex data |
|---|---|---|
| Stacked autoencoder (SAE) | 92.12 | 94.02 |
| Stacked sparse autoencoder | 93.41 | 94.92 |

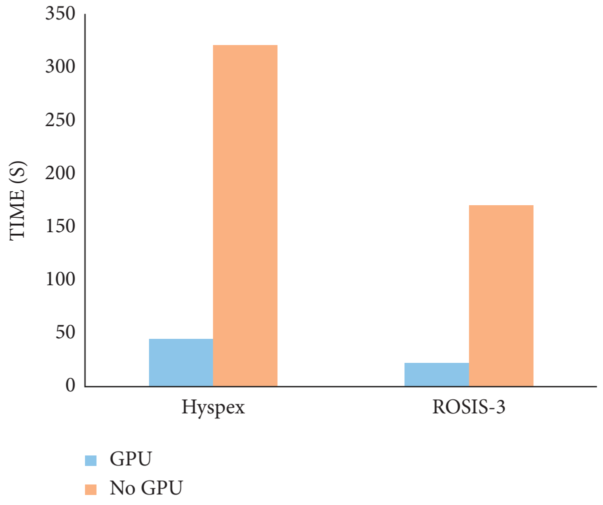


Figure 13: Comparison of classification speed under GPU parallel computing.

## 5. Conclusions

In this paper, the two models, SAE and DBN in deep learning, were applied to hyperspectral image classification to address some of the problems of hyperspectral image classification. The classification model was formed by connecting to a softmax regression classifier. Multiple experiments were conducted by changing the number of hidden layer neurons of the AE and the RBM. The reconstructed image features were visualized and analyzed in comparison with the original image features, and the best reconstructed effect was obtained with 30 hidden layer neurons for the AE and 30 hidden layer neurons for the RBM. The AE and RBM were compared with the traditional feature extraction algorithm, and we experimentally verified that both classification algorithms based on the AE and RBM outperformed the classification algorithm based on the traditional feature extraction algorithm in terms of classification performance; the AE was stacked to form an SAE, and the RBM was stacked to form a deep belief network; that is, the single-layer neural network was stacked to form a deep network. Through the experimental analysis, the classification accuracy reached 92.12% and 94.02% with three hidden layers of SAEs and 88.22% and 92.16% with two hidden layers of deep belief networks; thus, the optimal classification model was found to be an SAE. The classification accuracy was further improved to 93.41% and 94.92% by modifying the model to a stacked sparse autoencoder. To shorten the long training time of the deep learning algorithm, GPU computing was applied to the classification model, improving classification speed by more than seven times.

Through comprehensive experimental analysis, we found that the hyperspectral image classification model based on a deep learning algorithm can obtain better classification accuracy than the traditional classification model. However, some problems with the deep learning algorithm still need to be addressed. For example, the selection of model hyperparameters and the number of neurons and layers in the hidden layer have a large impact on the classification accuracy. That is, the selection of model parameters is complicated and has a large impact on the classification performance. Second, the deep learning algorithm is complicated, and under the same hardware equipment conditions, its performance varies considerably. How to improve the optimization algorithm accuracy while reducing its computing time will also need to be studied further.

## Data Availability

The data used to support the findings of this research are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant nos. 2021YFB3900505 and 2021YFC3000401), the National Natural Science Foundation of China (Grant no. 41941019), the Sichuan Mineral Resources Research Center (Grant no. SCKCZY2021-ZC003), the Sichuan Provincial Department of Education Humanities and Social Sciences (Zhang Daqian Research) Key Project (Grant no. ZDQ2021–01), the Open Foundation of Sichuan Center for Disaster Economic Research (Grant no. ZHJJ2021-ZD001), and the Industry-School Cooperative Education Program of the Ministry of Education (Grant nos. 202101162001 and 202102245035).

## References

[1] P. R. Jeyaraj, B. K. Panigrahi, and E. R. S. Nadar, “Classifier feature fusion using deep learning model for non-invasive detection of oral cancer from hyperspectral image,” *IETE Journal of Research*, vol. 68, no. 1, pp. 1–12, 2020.

[2] X. Li, B. Liu, K. Zhang et al., “Multi-view learning for hyperspectral image classification: an overview,” *Neurocomputing*, vol. 500, pp. 499–517, 2022.

[3] J. He and I. Barton, “Hyperspectral remote sensing for detecting geotechnical problems at Ray mine,” *Engineering Geology*, vol. 292, Article ID 106261, 2021.

[4] L. Liu, T. Zhang, F. Yin, H. Zhang, K. Cao, and J. Hong, “Remote sensing of geological features and mineral prospecting indicators for Pb-Zn deposits in the Khuzdar-Lasbela zone, Pakistan,” *Acta Geologica Sinica*, vol. 96, no. 3, pp. 1012–1025, 2022.

[5] S. K. Saha, “Remote sensing and geographic information system applications in hydrocarbon exploration: a review,” *Journal of the Indian Society of Remote Sensing*, vol. 50, no. 8, pp. 1457–1475, 2022.

[6] Z. Wang and S. Tian, “Ground object information extraction from hyperspectral remote sensing images using deep learning algorithm,” *Microprocessors and Microsystems*, vol. 87, Article ID 104394, 2021.

[7] C. Wu, X. Li, W. Chen, and X. Li, “A review of geological applications of high-spatial-resolution remote sensing data,” *Journal of Circuits, Systems, and Computers*, vol. 29, no. 6, Article ID 2030006, 2020.

[8] C. J. Legleiter, T. V. King, K. D. Carpenter et al., “Spectral mixture analysis for surveillance of harmful algal blooms (SMASH): a field-laboratory-and satellite-based approach to identifying cyanobacteria genera from remotely sensed data,” *Remote Sensing of Environment*, vol. 279, Article ID 113089, 2022.

[9] C. Zang, F. Shen, and Z. Yang, “Aquatic environmental monitoring of inland waters based on UAV hyperspectral remote sensing,” *Remote Sensing for Natural Resources*, vol. 33, no. 3, pp. 45–53, 2021.

[10] D. Zheng, S. Guo-chun, W. Bo-jian et al., “Classification of dominant species in coniferous and broad-leaved mixed forest on Changbai Mountain based on UAV-based hyperspectral image and deep learning algorithm,” *Chinese Journal of Ecology*, vol. 41, no. 5, pp. 1024–1032, 2022.

[11] Y. Le, M. Hu, Y. Chen et al., “Investigating the shallow-water bathymetric capability of zhuhai-1 spaceborne hyperspectral images based on ICESat-2 data and empirical approaches: a case study in the south China sea,” *Remote Sensing*, vol. 14, no. 14, p. 3406, 2022.

[12] P. Liu, H. Zhang, and K. B. Eom, “Active deep learning for classification of hyperspectral images,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 712–724, 2017.

[13] S. Liu and Q. Shi, “Multitask deep learning with spectral knowledge for hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 12, pp. 2110–2114, 2020.

[14] G. Sun, H. Fu, J. Ren et al., “SpaSSA: superpixelwise adaptive SSA for unsupervised spatial-spectral feature extraction in hyperspectral image,” *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6158–6169, 2022.

[15] B. M. Shahshahani and D. A. Landgrebe, “The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

[16] G. Sun, J. Cheng, A. Zhang, X. Jia, Y. Yao, and Z. Jiao, “Hierarchical fusion of optical and dual-polarized SAR on impervious surface mapping at city scale,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 264–278, 2022.

[17] U. Khan, S. Paheding, C. P. Elkin, and V. K. Devabhaktuni, “Trends in deep learning for medical hyperspectral image analysis,” *IEEE Access*, vol. 9, pp. 79534–79548, 2021.

[18] Z. Lei, Y. Zeng, P. Liu, and X. Su, “Active deep learning for hyperspectral image classification with uncertainty learning,”

*IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[19] F. Lu and M. Han, "Hyperspectral remote sensing image classification based on deep extreme learning machine," *Journal of Dalian University of Technolgy*, vol. 58, no. 2, pp. 166–173, 2018.

[20] L. Ma, G. Lu, D. Wang et al., "Deep learning based classification for head and neck cancer detection with hyperspectral imaging in an animal model," in *Proceedings of the SPIE Biomedical Applications in Molecular, Structural, and Functional Imaging Conference*, The Society of Photographic Instrumentation Engineers, Orlando, FL, USA, 2017.

[21] N. Falco, J. A. Benediktsson, and L. Bruzzone, "Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 6223–6240, 2015.

[22] C. Jayaprakash, B. B. Damodaran, and K. Soman, "Randomized ICA and LDA dimensionality reduction methods for hyperspectral image classification," *Journal of Applied Remote Sensing*, vol. 14, no. 3, 2018.

[23] H. Fu, G. Sun, J. Ren, A. Zhang, and X. Jia, "Fusion of PCA and segmented-PCA domain multiscale 2-D-SSA for effective spectral-spatial feature extraction and data classification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[24] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: a superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4581–4593, 2018.

[25] R. Lazcano, D. Madroñal, R. Salvador et al., "Porting a PCA-based hyperspectral image dimensionality reduction algorithm for brain cancer detection on a manycore architecture," *Journal of Systems Architecture*, vol. 77, pp. 101–111, 2017.

[26] Q. Hou, Y. Wang, L. Jing, and H. Chen, "Linear discriminant analysis based on kernel-based possibilistic c-means for hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 8, pp. 1259–1263, 2019.

[27] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 5, pp. 894–898, 2011.

[28] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[29] D. Cai, X. He, and J. Han, "Efficient kernel discriminant analysis via spectral regression," in *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 427–432, IEEE, Orlando, FL, USA, October 2007.

[30] D. Cai, X. He, and J. Han, "Speed up kernel discriminant analysis," *The VLDB Journal*, vol. 20, no. 1, pp. 21–33, 2011.

[31] K. Zhang, W. Song, H. Shi, and Q. Zhou, "Fault detection based on improved kernel principal component analysis," *Control Engineering China*, vol. 24, no. 2, pp. 418–424, 2017.

[32] B. Liu, X. Yu, A. Yu, and G. Wan, "Deep convolutional recurrent neural network with transfer learning for hyperspectral image classification," *Journal of Applied Remote Sensing*, vol. 12, no. 2, p. 1, 2018.

[33] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7758–7772, 2021.

[34] K. Safari, S. Prasad, and D. Labate, "A multiscale deep learning approach for high-resolution hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 1, pp. 167–171, 2021.

[35] S. Singh and S. S. Kasana, "Efficient classification of the hyperspectral images using deep learning," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27061–27074, 2018.

[36] X. Zhou, S. Li, F. Tang, K. Qin, S. Hu, and S. Liu, "Deep learning with grouped features for spatial spectral classification of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 97–101, 2017.

[37] N. Audebert, B. Le Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: a comparative review," *IEEE Geoscience And Remote Sensing Magazine*, vol. 7, no. 2, pp. 159–173, 2019.

[38] C. Li, Y. Wang, X. Zhang, H. Gao, Y. Yang, and J. Wang, "Deep belief network for spectral-spatial classification of hyperspectral remote sensor data," *Sensors*, vol. 19, no. 1, p. 204, 2019.

[39] H. Fu, A. Zhang, G. Sun et al., "A novel band selection and spatial noise reduction method for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[40] H. Li, W. Y. Wang, L. Pan, W. Li, Q. Du, and R. Tao, "Robust capsule network based on maximum correntropy criterion for hyperspectral image classification," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 738–751, 2020.

[41] J. Li, B. Xi, Q. Du, R. Song, Y. Li, and G. Ren, "Deep kernel extreme-learning machine for the spectral-spatial classification of hyperspectral imagery," *Remote Sensing*, vol. 10, no. 12, p. 2036, 2018.

[42] J. Li, B. Xi, Y. Li, Q. Du, and K. Wang, "Hyperspectral classification based on texture feature enhancement and deep belief networks," *Remote Sensing*, vol. 10, no. 3, p. 396, 2018.

[43] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: an overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.

[44] Y. Dong, C. Yang, and Y. Zhang, "Deep metric learning with online hard mining for hyperspectral classification," *Remote Sensing*, vol. 13, no. 7, p. 1368, 2021.

[45] Z. Li, X. Tang, W. Li, C. Wang, C. Liu, and J. He, "A two-stage deep domain adaptation method for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 7, p. 1054, 2020.

[46] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, and L. Tang, "Hyperspectral images classification with gabor filtering and convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2355–2359, 2017.

[47] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[48] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sensing*, vol. 12, no. 6, p. 923, 2020.

[49] X. Sun, F. Zhou, J. Dong, F. Gao, Q. Mu, and X. Wang, "Encoding spectral and spatial context information for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2250–2254, 2017.

[50] J. Zabalza, J. Ren, J. Zheng et al., "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[51] J. Guo, Y. Li, S. Dong, and W. Zhang, "Innovative method of crop classification for hyperspectral images combining stacked autoencoder and CNN," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 52, no. 12, pp. 225–232, 2021.

[52] G. Sun, X. Zhang, X. Jia et al., "Deep fusion of localized spectral features and multi-scale spatial features for effective classification of hyperspectral images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 91, Article ID 102157, 2020.

[53] H. Madani and K. McIsaac, "Distance transform-based spectral-spatial feature vector for hyperspectral image classification with stacked autoencoder," *Remote Sensing*, vol. 13, no. 9, p. 1732, 2021.

[54] H. Sun, J. Ren, H. Zhao, P. Yuen, and J. Tschannerl, "Novel gumbel-softmax trick enabled concrete autoencoder with entropy constraints for unsupervised hyperspectral band selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[55] K. Hanbay, "Evrişimsel sinir ağı ve iki-boyutlu karmaşık gabor dönüşümü kullanılarak hiperspektral görüntü sınıflandırma," *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, vol. 35, no. 1, pp. 443–456, 2019.

[56] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1424–1436, 2021.

[57] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[58] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1259–1270, 2018.

[59] Y. Xu, B. Du, and L. Zhang, "Self-attention context network: addressing the threat of adversarial attacks for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 8671–8685, 2021.

[60] X. Dai, J. Cheng, Y. Gao et al., "Deep belief network for feature extraction of urban artificial targets," *Mathematical Problems in Engineering*, vol. 2020, no. 10, Article ID 2387823, 13 pages, 2020.

[61] H. Gong, T. Mu, Q. Li et al., "Hyperspectral image classification using 3D-2D CNN with multi-scale information extraction and fusion module," in *Proceedings of the 4th International Conference on Photonics and Optical Engineering*, vol. 11761, Xian, China, November 2020.

[62] Z. Gong, W. Hu, X. Du, P. Zhong, and P. Hu, "Deep manifold embedding for hyperspectral image classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10430–10443, 2022.

[63] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3599–3618, 2019.

[64] X. Hu, W. Yang, H. Wen, Y. Liu, and Y. Peng, "A lightweight 1-D convolution augmented transformer with metric learning for hyperspectral image classification," *Sensors*, vol. 21, no. 5, p. 1751, 2021.

[65] J. Mi, M. Zhang, Z. Zhu et al., "Morphological wave attenuation of the nature-based flood defense: a case study from Chongming Dongtan Shoal, China," *Science of the Total Environment*, vol. 831, Article ID 154813, 2022.

[66] M. Zhang, Z. Dai, T. J. Bouma et al., "Tidal-flat reclamation aggravates potential risk from storm impacts," *Coastal Engineering*, vol. 166, no. 9, Article ID 103868, 2021.

[67] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.

[68] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 113, pp. 155–165, 2016.