

## Research Article

# Quasiconformal Mapping Kernel Machine Learning-Based Intelligent Hyperspectral Data Classification for Internet Information Retrieval

Jing Liu and Yulong Qiao 

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Yulong Qiao; [qiaoyulonghrbeu@126.com](mailto:qiaoyulonghrbeu@126.com)

Received 2 May 2020; Revised 13 July 2020; Accepted 20 August 2020; Published 1 September 2020

Academic Editor: Pei-Wei Tsai

Copyright © 2020 Jing Liu and Yulong Qiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent internet data mining is an important application of AIoT (Artificial Intelligence of Things), and it is necessary to construct large training samples with the data from the internet, including images, videos, and other information. Among them, a hyperspectral database is also necessary for image processing and machine learning. The internet environment provides abundant hyperspectral data resources, but the hyperspectral data have no class labels and no so high value for applications. So, it is important to label the class information for these hyperspectral data through machine learning-based classification. In this paper, we present a quasiconformal mapping kernel machine learning-based intelligent hyperspectral data classification algorithm for internet-based hyperspectral data retrieval. The contributions include three points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification, the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Experiments show that the proposed scheme is effective for hyperspectral image classification and retrieval.

## 1. Introduction

Intelligent data mining is an important issue of AIoT (Artificial Intelligence of Things), and with the development of machine learning, a large training dataset is necessary for the learning tasks, including images and videos. Among these applications, hyperspectral databases are also very necessary for hyperspectral image processing and machine learning. So, internet environment-based hyperspectral data retrieval is an important issue of AIoT, and it is also an effective way to create a large-scale hyperspectral training database for some applications. The internet environment provides abundant hyperspectral data resources but is included in other complex data. Moreover, the hyperspectral data have no class labels and no detail class knowledge information for these data. So, the hyperspectral data have no high value without class information. So, further intelligent hyperspectral data-

based machine learning is necessary through internet-based data retrieval. Intelligent hyperspectral data retrieval under the internet environment combination is the application of AI (Artificial Intelligence) and IoT (Internet of Things).

Hyperspectral data-based machine learning is a feasible and effective method to extract the features for image retrieval. The machine learning methods are divided into unsupervised and supervised learning. The unsupervised learning includes multidimensional scaling, NMF, ICA, neighborhood preserving embedding, Locality Preserving Projection (LPP) [1], and other computing methods [2], and for the supervised learning, generalized discriminant analysis [3], uncorrelated discriminant vector analysis [4], and some acceleration algorithm [5, 6]. In recent years, the kernel-based machine learning algorithms were presented for the feature extraction; this paper proposes an improved kernel function supervised kernel-based LPP, local structure

supervised feature extraction [7], kernel subspace LDA [8], kernel MSE [9], and quasiconformal mapping-based kernel machine [10]. Many kernel learning methods are proposed to improve the accuracy performance for the practical kernel learning system, for example, sparse multiple kernel learning [11], large scale multiple kernel learning [12], and Lp-norm multiple kernel learning [13]. With the development of the deep learning theory, the framework of the deep learning-based multikernel machine is an effective framework, and the learning method has been widely used in image analysis [14, 15], image annotation [16], image classification [17], image segmentation [18], and anomaly detection [19]. Researchers proposed deep kernel learning, namely, LMKL [20], and in the other work, the estimated value of missing an error is adjusted instead of the double objective function [21]. And recent research includes machine learning-based image processing [22, 23] under the internet environment for the application of AI and IoT. Mathematically, it has been proved that multilayer can improve the richness of representation, and the researchers combine the support vector machine and multiple classifiers and use an adaptive back propagation algorithm to update coefficients and weights [20, 26, 27].

The kernel-based machine learning on the hyperspectral data is proposed to retrieve the spectral data in the internet environment. In the algorithm, we proposed the quasiconformal mapping-based kernel learning for hyperspectral data classification for data retrieval in the internet environment. And the Mahalanobis distance kernel function is applied to extract the nonlinear feature, with higher discriminative ability than Euclidean distance-based kernel function learning. The objective function of quasiconformal kernel learning created with the Fisher criterion is proposed to seek the optimal parameters of the quasiconformal mapping projection. The proposed scheme is effective to hyperspectral image retrieval under the internet environment.

## 2. Proposed Algorithm

*2.1. Motivation and Framework.* Intelligent hyperspectral retrieval extracts the features of the spectrum through sensing data processing and analysis. Motivated by the fact that kernel machine-based spectrum learning is effective to the nonlinear classification, we present a framework of quasiconformal mapping-based multiple kernel learning with Mahalanobis distance kernel functions. The contributions include three points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification; the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning; and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. The performance is improved with two facts: one is to optimize the data structure in the kernel empirical space with quasiconformal mapping and second is to improve the discriminant ability with Mahalanobis distance-based kernel. The proposed algo-

gorithm has the highly effective performance on characterizing the data through solving complex visual learning tasks. The learning framework of hyperspectral image classification is presented in Figure 1.

*2.2. Quasiconformal Kernel Mapping Learning.* Kernel-based learning is used in data classification, with “empirical kernel map.” Suppose that matrix  $K = [k_{ij}]_{m \times m}$ ,  $K$  is decomposed as

$$K_{m \times m} = P_{m \times r} \Lambda_{r \times r} P_{r \times m}^T, \quad (1)$$

where  $\Lambda$  is a diagonal matrix with  $r$  positive eigenvalues of  $K$ ,  $P$ . The mapping is the empirical kernel map  $\Phi_r^e \in R^r$  as

$$\begin{aligned} \Phi_r^e : \chi &\longrightarrow R^r, \\ x &\longrightarrow \Lambda^{-1/2} P^T (k(x, x_1), k(x, x_2), \dots, k(x, x_m)). \end{aligned} \quad (2)$$

Different kernels have different abilities on classification, and the kernel function based on feature similarity refers that features exist in the form of distance, that is, similarity in a function expression. For sample features  $x, y \in R^N$ , if the Euclidean distance between the two is  $d_U(x, y) = \|x - y\|$ , the general form of such a kernel function is

$$k^U(x, y) = f(d_U(x, y)). \quad (3)$$

The RBF kernel is the most typical representative of this kind of kernel function, in addition to the negative distance kernel, the logarithmic function kernel, and the Bn spline kernel. For this type of kernel function, the Euclidean distance  $d_U(x, y)$  can be simply replaced with the Mahalanobis distance  $d_M(x, y) = \sqrt{(x - y)^T \mathbf{M} (x - y)}$ , that is,

$$k_M^U(x, y) = f(d_M(x, y)). \quad (4)$$

In particular, the expression for a typical Mahalanobis distance RBF kernel function is

$$k_M^{\text{RBF}}(x, y) = \exp \left( -\frac{1}{2\sigma^2} (x - y)^T \mathbf{M} (x - y) \right). \quad (5)$$

Similar to the above expression, given the inner product  $I(x, y) = x \cdot y$  of two features, the general form of the kernel function based on the feature inner product is

$$k^I(x, y) = f(I(x, y)). \quad (6)$$

The polynomial kernel is the most typical representative of this kind of kernel function, in addition to the Sigmoid kernel. For this type of kernel function, a transformation is required when expanding to the Mahalanobis distance kernel function. Considering that in the case of Euclidean distance, the inner product of the two features satisfies

$$I(x, y) = x \cdot y = \frac{1}{2} \cdot (\|x - 0\|^2 + \|y - 0\|^2 - \|x - y\|^2). \quad (7)$$

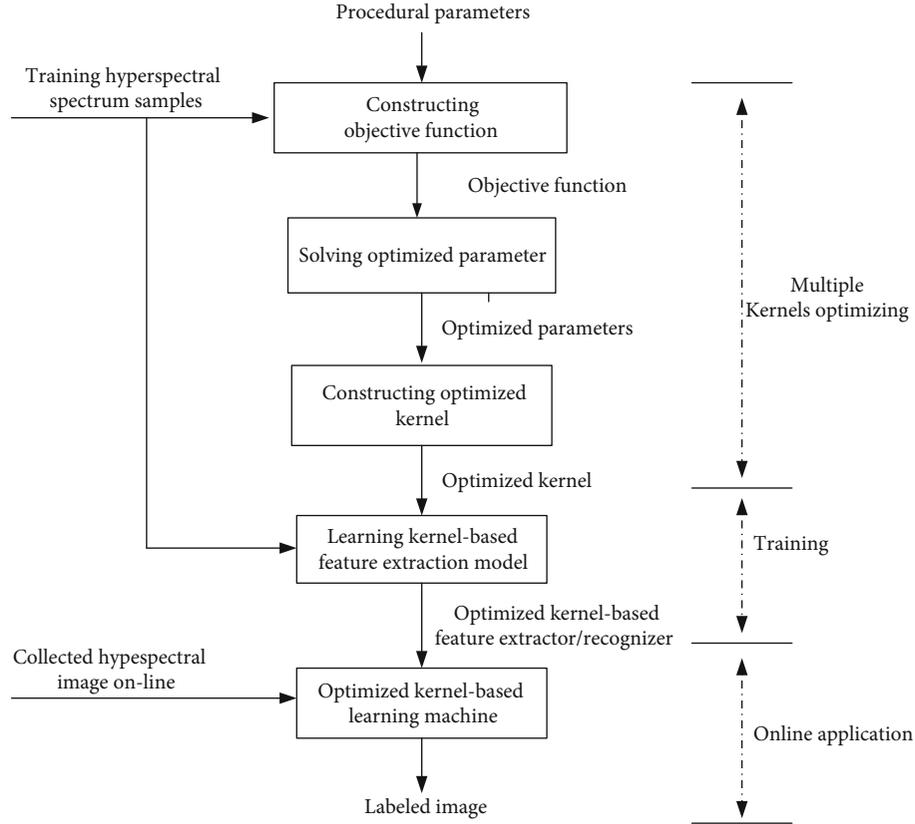


FIGURE 1: Algorithm framework.

Therefore, the Mahalanobis distance of the inner product can be written as

$$d_M^l(x, y) = \frac{1}{2} (d_M^2(x, 0) + d_M^2(y, 0) - d_M^2(x, y)). \quad (8)$$

In this way, the Mahalanobis distance of the inner product can be obtained by calculating the Mahalanobis distance between the feature and the origin 0 and the feature. Similarly, the Mahalanobis distance of the inner product can be obtained:

$$k_M^l(x, y) = f(d_M^l(x, y)). \quad (9)$$

In particular, the expression for a typical Mahalanobis distance polynomial kernel function is

$$k_M^{\text{POL}}(x, y) = \left( \frac{1}{2} (x^T \mathbf{M}x + y^T \mathbf{M}y - (x - y)^T \mathbf{M}(x - y)) + c \right)^d. \quad (10)$$

In this paper, we introduce the quasiconformal kernel,  $k_q(x, y)$ , as follows:

$$k_q(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})f(\mathbf{y})k(\mathbf{x}, \mathbf{y}), \quad (11)$$

where  $\mathbf{x}, \mathbf{y}$  are the sample vectors and  $f(\mathbf{x})$  is

$$f(\mathbf{x}) = b_0 + \sum_{n=1}^{N_{XV}} b_n e(\mathbf{x}, \tilde{\mathbf{x}}_n), \quad (12)$$

where  $e(\mathbf{x}, \tilde{\mathbf{x}}_n) = e^{-\delta \|\mathbf{x} - \tilde{\mathbf{x}}_n\|^2}$ ,  $\delta$  is a free parameter,  $\tilde{\mathbf{x}}_n$  is called the ‘‘expansion vectors (XVs)’’,  $N_{XV}$  is the number of XVs, and  $b_n$  ( $n = 0, 1, 2, \dots, N_{XV}$ ) is the ‘‘expansion coefficients’’ associated with  $\tilde{\mathbf{x}}_n$  ( $n = 0, 1, 2, \dots, N_{XV}$ ).

On the multiple kernels, the quasiconformal mapping kernel is described as

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) \sum_{i=1}^m a_i k_{0,i}(\mathbf{x}, \mathbf{x}') f(\mathbf{x}'), \quad (13)$$

where  $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$ ,  $k_{0,i}(\mathbf{x}, \mathbf{x}')$  is the  $i$ th basic kernel,  $m$  is the number of basic kernels for combination,  $a_i \geq 0$  is the weight for the  $i$ th basic kernel function, and  $q(\cdot)$  is the factor function defined by

$$f(\mathbf{x}) = b_0 + \sum_{i=1}^n b_i k_0(\mathbf{x}, \mathbf{a}_i), \quad (14)$$

where  $k_0(\mathbf{x}, \mathbf{a}_i) = e^{-\gamma \|\mathbf{x} - \mathbf{a}_i\|^2}$ ,  $\mathbf{a}_i \in \mathbb{R}^d \{\mathbf{a}_i, \forall i = 1, 2, \dots, n\}$  are selected by the training samples, and  $b_i$  is the coefficient for the combination.  $k(\mathbf{x}, \mathbf{x}')$  satisfies the Mercer condition,  $k(\mathbf{x}$

,  $\mathbf{x}'$ ) is rewritten as  $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^m d_i [f(\mathbf{x})k_{0,i}(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')] /$  optimized transformation  $k_{0,i}(\mathbf{x}, \mathbf{x}')$ , and  $q(\mathbf{x})k_{0,i}(\mathbf{x}, \mathbf{x}')q(\mathbf{x}')$  is the linear combination of kernels.

Given  $\mathbf{d} = [d_1, d_2, \dots, d_m]$ ,  $\alpha = [\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_n]$ , and the optimized multiple kernels  $k(x, x') = K_f(k_{0,i}(x, x'), \mathbf{d}, \alpha)$ ,  $\mathbf{d}$ ,  $\alpha$  are to be optimized for the classification task. Finally, the joint convex formulation can be formed as

$$\begin{aligned} & \max_{\mathbf{d}, \alpha} F_c(K_0, \mathbf{d}, \alpha) \\ & \text{Subject to } \|\mathbf{d}\| = 1, \|\alpha\| = 1. \end{aligned} \quad (15)$$

$F_c(\cdot)$  measures the class discriminative ability.  $\mathbf{d}$ ,  $\alpha$  can be solved in two stages, respectively. In the first stage, the centered kernel alignment [28] is applied to define the objective optimization function; in the second stage, Fisher-based and Margin-based optimization function is used to solve  $\alpha$ .

*Step 1.* Optimize the weight vector of kernels.

In multikernel learning, the crucial step is to select the adaptive weights of multiple kernels. The weight  $\mathbf{d} = [d_1, d_2, \dots, d_m]$  is to solve with centered kernel alignment [22]:

$$\begin{aligned} & \max O_c(K_0^{(C)}, K^*) \\ & \text{Subject to } K_0^{(C)} = \sum_{i=1}^m d_i K_{0,i}^{(C)}, \text{tr}(K_0) = 1, d_i \geq 0, \forall i, \end{aligned} \quad (16)$$

where  $O_c(K_0^{(C)}, K^*)$  is the objective function,

$$K^*(x, x') = \begin{cases} 1, & \text{if } y = y', \\ \frac{-1}{c-1}, & \text{if } y \neq y', \end{cases} \quad (17)$$

is the optimal kernel, and  $\text{tr}$  is the matrix trace.  $K_0^{(C)} = [I - 11^T/m]K_0[I - 11^T/m]$  is the centered kernel matrix of  $K_0$ ,  $I$  is the identity matrix, and  $\mathbf{1}$  is a vector with all entries equal to 1. Accordingly,  $K_{0,i}^{(C)} = [I - 11^T/m]K_{0,i}[I - 11^T/m]$  is the center of  $K_{0,i}$ ,  $i = 1, 2, \dots, m$ . The objective function  $O_c(K_0^{(C)}, K^*) = \langle K_0^{(C)}, K^* \rangle_F / \|K_C^*\|_F \|K_0^{(C)}\|_F$ ,  $K_C^*$  is the centered kernel matrix of  $K^*$ , where  $\langle \cdot, \cdot \rangle_F$  is the Frobenius norm between two matrices, i.e.,  $\langle D, E \rangle_F = \sum_{i=1}^m \sum_{j=1}^m d_{ij} e_{ij} = \text{tr}(DE^T)$ . Suppose  $\mathbf{d} = [d_1, d_2, \dots, d_m]$ ,  $\mathbf{d}^*$  is solved with the quadratic programming (QP) problem:

$$\begin{aligned} & \min v^T \mathbf{T} v - 2v^T \boldsymbol{\eta} \\ & \text{Subject to } v_i \geq 0, \forall i, \end{aligned} \quad (18)$$

where  $\mathbf{d}^* = v^* / \|\mathbf{d}^*\|_2$ ,  $\boldsymbol{\eta} = [\langle K_{0,1}^{(C)}, K^* \rangle_F, \langle K_{0,2}^{(C)}, K^* \rangle_F, \dots, \langle K_{0,m}^{(C)}, K^* \rangle_F]^T$ , and  $\mathbf{T}$  is defined by  $\mathbf{T}_{ij} = \langle K_{0,i}^{(C)}, K_{0,j}^{(C)} \rangle_F$ ,  $i, j = 1, 2, \dots, m$ . Based on this centered kernel alignment, the optimization problem can be trans-

formed into a QP problem, which can be effectively solved with OPTI toolbox [28].

*Step 2.* Optimize the coefficients of kernels.

The coefficients  $\alpha = [b_1, b_2, \dots, b_n]$  is solved based on the Fisher criterion and  $J_{\text{Fisher}}(\alpha)$  is defined as

$$J_{\text{Fisher}}(\alpha) = \cdot \quad (19)$$

Then,  $J_{\text{Fisher}}(\alpha) / \partial \alpha = 2/J_2^2 (J_2 \mathbf{E}^T \mathbf{B}_0 \mathbf{E} - \mathbf{J}_1 \mathbf{E}^T \mathbf{W}_0 \mathbf{E}) \alpha$ , where  $J_{\text{Fisher}}$  is solved with the eigenvalue problem of  $(\mathbf{E}^T \mathbf{W}_0 \mathbf{E})^{-1} (\mathbf{E}^T \mathbf{B}_0 \mathbf{E})$  and  $\alpha$ . The matrix  $(\mathbf{E}^T \mathbf{W}_0 \mathbf{E})^{-1} (\mathbf{E}^T \mathbf{B}_0 \mathbf{E})$  maybe is not symmetrical or the matrix  $\mathbf{E}^T \mathbf{W} \mathbf{E}$  is singular;  $\alpha$  is solved as

$$\alpha^{(n+1)} = \alpha^{(n)} + \varepsilon \left( \frac{1}{J_2} \mathbf{E}^T \mathbf{B}_0 \mathbf{E} - \frac{J_{\text{Fisher}}}{J_2} \mathbf{E}^T \mathbf{W}_0 \mathbf{E} \right) \alpha^{(n)}, \quad (20)$$

where  $\varepsilon$  is the learning rate,  $\varepsilon(n) = \varepsilon_0 (1 - n/N)$  is the learning rate of the  $n$ th iteration,  $\varepsilon_0$  is the initial learning rate, and  $N$  is the number of the total iterations.

### 2.3. Metric Similarity-Based Learning

*2.3.1. Similar/Dissimilar Function.* Suppose the initial distance metric function  $d(x, y)$  (generally using Euclidean distance), the purpose of metric learning is to construct a new distance  $\tilde{d}(x, y)$  based on some prior information, which is more consistent with the description of the sample features than the initial distance metric. In order to achieve this, the new metric function  $\tilde{d}(x, y)$  can be converted into  $d(f(x), f(y))$ , that is, by defining a mapping  $f$  and using the original distance metric function to calculate a new metric function, thereby converting the metric learning problem into a learning mapping function  $f$  problem.

Given a sample set  $X \in R^N$ ,  $x, y, z$  is a sample in the sample set; if a function  $d : X \times X \rightarrow R^+$  defined in the vector space satisfies the following properties, then  $d$  is called a distance measure function: symmetry:  $d(x, y) = d(y, x)$ , non-negative:  $d(x, y) \geq 0$ , distinguishability:  $d(x, y) = 0 \Leftrightarrow x = y$ , and triangle inequality:  $d(x, y) + d(y, z) \geq d(x, z)$ . In the metric function that satisfies the above properties, the Euclidean distance is the most common distance metric function, which measures the absolute distance between spatial sample points, which is defined as

$$d(x, y) = \sqrt{(x - y)^T (x - y)}. \quad (21)$$

The Euclidean distance is characterized by simple calculation, but since the absolute distance measured is directly related to the coordinates of the position of each point, the adaptability to the data is poor in terms of feature scale and coupling degree between features. Cosine similarity mainly measures the consistency of direction between two vectors, which is defined as

$$d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}. \quad (22)$$

Compared with the Euclidean distance, the cosine similarity measures the angle between the space vectors, which reflects the difference in the direction of the vector and is insensitive to the absolute value.

The Minkowski distance is a general expression of a class of distance functions. Given two  $N$ -dimensional vectors:  $x = \{x_1, x_2, \dots, x_N\}$  and  $y = \{y_1, y_2, \dots, y_N\}$ , the Minkowski distance is defined as

$$d(x, y) = \left( \sum_{s=1}^N (x_s - y_s)^P \right)^{1/P}, \quad (23)$$

where  $P$  takes a different value; the distance is derived as a different type of distance. When  $P = 1$ , the distance is called Manhattan distance, that is,  $d(x, y) = (\sum_{s=1}^N |x_s - y_s|)$ . This distance is used to indicate the absolute wheelbase sum of the two points on the standard coordinate system. When  $P = 2$ , the distance is called the Euclidean distance. When  $P \rightarrow \infty$ , the Minkowski distance is called the Chebyshev distance, that is,  $d(x, y) = \max_s (|x_s - y_s|)$ , which represents the maximum value of the numerical difference between the coordinates. Similar to the Euclidean distance, the Minkowski distance is still related to the dimension of the feature, and the correlation between the features is not considered.

The Mahalanobis distance was proposed by P.C. Mahalanobis. It is defined as

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} = \sqrt{(x - y)^T \mathbf{M} (x - y)}, \quad (24)$$

where  $S$  is the data correlation matrix and  $\mathbf{M} = S^{-1}$  is the Mahalanobis matrix. When  $M$  is a unit array, the Mahalanobis distance degenerates into a Euclidean distance, indicating that the Euclidean distance is a special case of the Mahalanobis distance. The greatest advantage of the Mahalanobis distance is the ability to remove the coupling between various features and is scale-invariant.

As can be seen from the above definition, the traditional distance measurement function is fixed in form and can be calculated directly according to the formula without a process of "learning" to the sample. Obviously, this approach does not meet the diverse task requirements nor does it make full use of the information contained in the sample features. Therefore, it is necessary to construct a suitable distance metric function by learning the multidimensional information provided by the sample features for specific problems, so as to provide the best expression of feature similarity.

**2.3.2. Distance Measure Function.** Linear metric learning refers to obtaining a new metric function by linear transformation; that is, the form of the mapping function  $f$  is  $f(x) = \mathbf{A}^T x$ , where  $A$  is a projection matrix. The purpose of learning is to be able to find a suitable matrix  $A$ . In the case where the initial distance measure is Euclidean distance, the new measure function is

$$\tilde{d}(x, y) = \sqrt{(\mathbf{A}^T x - \mathbf{A}^T y)^T (\mathbf{A}^T x - \mathbf{A}^T y)}. \quad (25)$$

At this time, for the real matrix  $A$ , if  $\mathbf{M} = \mathbf{A}\mathbf{A}^T$ ,  $M$  is a semipositive symmetric matrix. The formula can be written as follows:

$$\begin{aligned} \tilde{d}(x, y) &= \sqrt{(\mathbf{A}^T x - \mathbf{A}^T y)^T (\mathbf{A}^T x - \mathbf{A}^T y)} \\ &= \sqrt{(x - y)^T \mathbf{A}\mathbf{A}^T (x - y)} \\ &= \sqrt{(x - y)^T \mathbf{M} (x - y)}. \end{aligned} \quad (26)$$

It can be seen from the above equation that under the effect of matrix  $A$ , the sample is linearly mapped to a new feature space. When  $A$  is a unit matrix, the distance is the Euclidean distance. When  $A$  is a diagonal matrix, it means that the original space is scaled, which is equivalent to weighting the samples. When  $A$  is an orthogonal matrix, it means that the original space is rotated and transformed. When  $A$  is a normal square matrix, it means that the original space is simultaneously scaled and rotated. When  $A$  is not a square matrix, it is equivalent to a dimensionality reduction operation in addition to rotation and scale transformation.

The above formula is similar to the Mahalanobis distance metric in form, but the traditional Mahalanobis distance metric uses the inverse of the covariance matrix as the Mahalanobis matrix.  $M$  in the above equation expands to the semipositive array, representing a more general Mahalanobis matrix, so linear metric learning is also known as Mahalanobis metric learning. From the perspective of using sample information, the traditional Mahalanobis matrix only uses the internal structure of the data, focusing on describing the distribution properties of the data, while the Mahalanobis matrix obtained by the metric learning makes full use of the relationship between the feature and the category label and focuses on features that adequately reflect sample class differences to achieve a better metric function. In general, a metric learning problem can be transformed into a constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{M}} \mathcal{L}_X(\mathbf{M}) + \lambda r(\mathbf{M}) \\ \text{s.t. } c_X(\mathbf{M}), \end{aligned} \quad (27)$$

where  $\mathcal{L}_X(\mathbf{M})$  represents the loss function on the training set  $X$ ;  $r(\mathbf{M})$  is a regularization term, which is used to correct the overfitting; and  $\lambda$  is the preset regularization factor, which is used to adjust the influence degree of the regularization term in the training process.  $c_X(\mathbf{M})$  is the constraint on the training set. Different learning algorithms can be derived depending on the difference of loss function, the regularization term, and the constraints.

**2.4. Similar/Dissimilar Learning Criteria.** Two popular criteria, Fisher criterion and large margin nearest neighbor criterion, are used in similar-/dissimilar-based learning.

**2.4.1. Fisher Criterion.** The method is based on the pairwise constraint information provided by the sample as a priori information to minimize the similar sample pairs and control the distance between the nonsimilar sample pairs as the basic idea to construct a convex optimization problem and achieve the purpose of learning the Mahalanobis distance matrix.

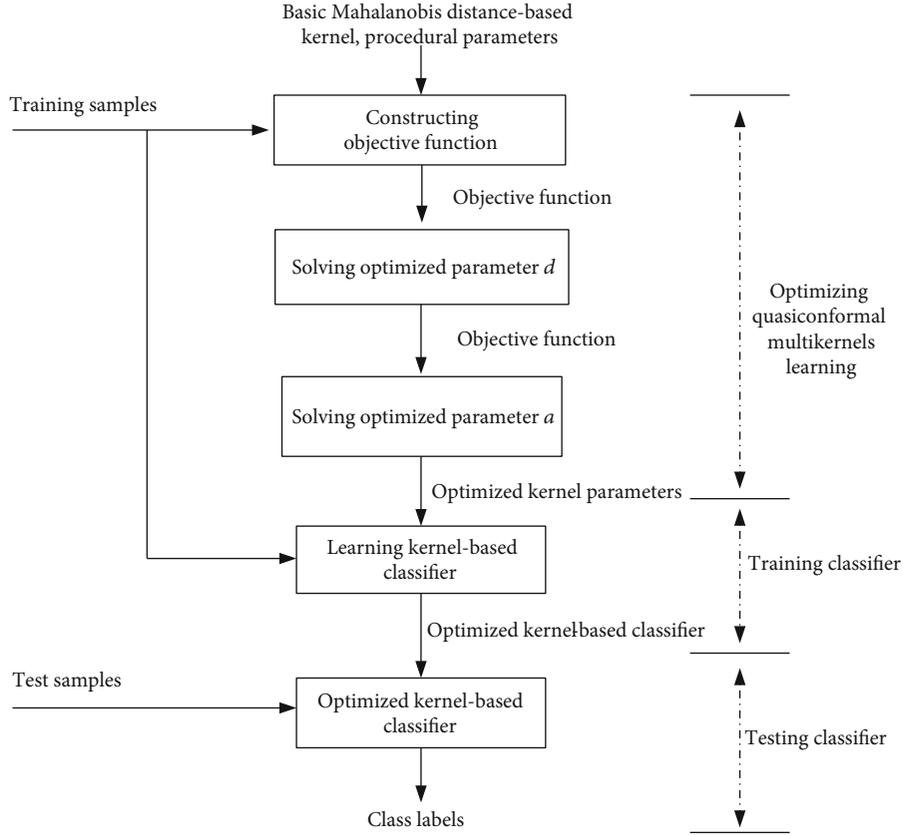


FIGURE 2: Data preprocessing procedure.

First, given a sample set  $X : \{x_i\}_{i=1}^M \subset \mathbb{R}^N$ , depending on whether the sample pairs belong to the same category, two constraint sets can be obtained: a homogeneous constraint set  $W$  and a nonlike constraint set  $B$ . If the categories of the sample pairs are the same (similar), then the sample pair belongs to the set  $W$ , and if the categories of the sample pairs are not the same (similar), the sample pair belongs to the set  $B$ . If the Euclidean distance is used as the initial distance metric, considering that the postlearning metric can make the distance between the pairs of similar samples as small as possible, a convex optimization problem can be constructed:

$$\begin{aligned} \min_{M \geq 0} \quad & \sum_{(x_i, x_j) \in W} d_M^2(x_i, x_j) \\ \text{s.t.} \quad & \sum_{(x_i, x_j) \in B} d_M(x_i, x_j) \geq 1, \end{aligned} \quad (28)$$

where  $d_M^2(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$  and  $M$  are semipositive definite matrices. The constraint is added mainly to remove the trivial solution of  $M = 0$ . In the specific solution, an iterative update method can be used to solve. In each iteration, the mature Newton downhill method is used to perform the gradient descent process to obtain an updated Mahalanobis matrix. The matrix is then iteratively mapped onto the constraint set. Although the algorithm is relatively simple in implementation, the corresponding calculation

amount is large in the case of large data size because the algorithm needs to construct all pairs of similar samples and non-similar samples in the whole dataset. And the convergence speed of the algorithm is also slow.

First, by introducing the projection matrix  $A$ , the distance between the pair of points becomes

$$d_M(x_i, x_j) = \sqrt{(\mathbf{A}^T x_i - \mathbf{A}^T x_j)^T (\mathbf{A}^T x_i - \mathbf{A}^T x_j)}. \quad (29)$$

Considering the constraint set  $W$ , after the action of the projection matrix  $A$ , the sum of the squares of the distances between all pairs of points is

$$d_W = \sum_{(x_i, x_j) \in W} (\mathbf{A}^T x_i - \mathbf{A}^T x_j)^T (\mathbf{A}^T x_i - \mathbf{A}^T x_j). \quad (30)$$

The sum of the squares of the distances between all pairs of points in constraint set  $B$  can be calculated as

$$d_B = \sum_{(x_i, x_j) \in B} (\mathbf{A}^T x_i - \mathbf{A}^T x_j)^T (\mathbf{A}^T x_i - \mathbf{A}^T x_j) = \text{tr}(\mathbf{A}^T \mathbf{S}_B \mathbf{A}). \quad (31)$$

Considering an excellent projection matrix  $A$ , it should maximize the distance between the samples in the constraint set  $B$  and reduce the distance between the samples in the

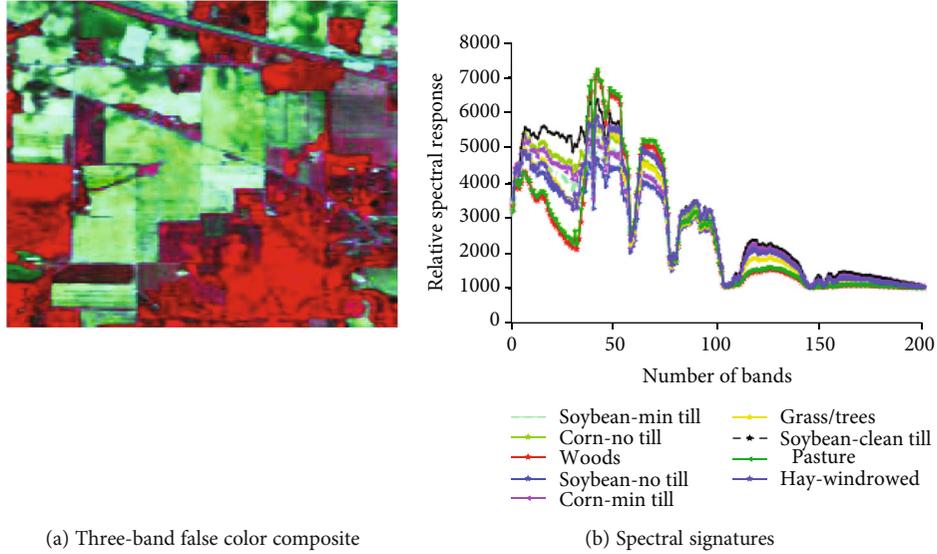


FIGURE 3: One example of Indian Pines data.

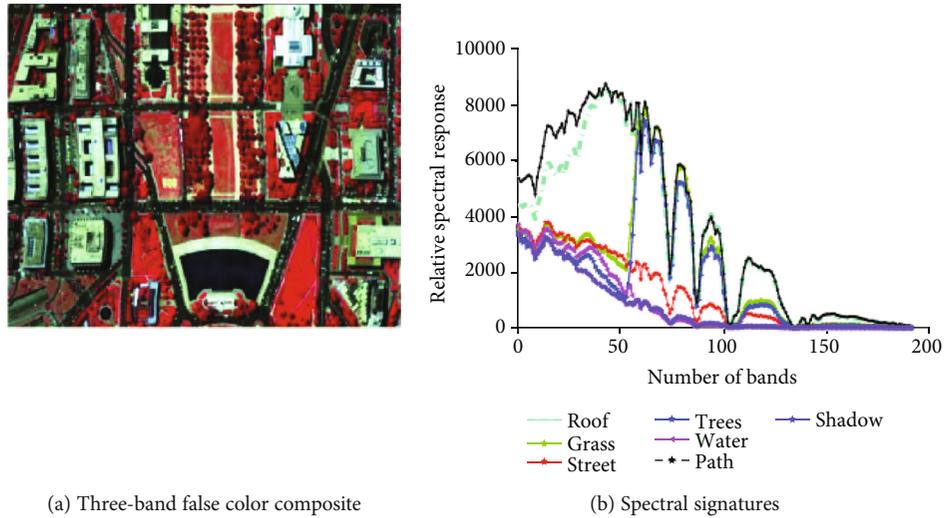


FIGURE 4: One example from D.C. Mall data.

TABLE 1: Performance on the Indian Pines data (%).

Class	1	2	3	4	5	6	7	8	9	10	11	12
SVC (polynomial)	49.32	58.73	96.45	39.26	65.82	93.65	62.92	85.33	99.01	65.83	72.33	58.41
SVC (Gaussian)	78.02	73.65	99.16	76.92	80.52	97.12	79.78	89.80	99.79	83.64	86.04	80.74
KSRC (polynomial)	51.83	59.68	96.13	49.12	78.56	93.87	62.83	84.72	98.23	67.57	75.27	60.77
KSRC (Gaussian)	77.84	76.47	99.12	75.56	79.06	97.42	82.71	88.73	98.69	83.93	86.38	81.12
SVC (Q-kernel)	78.32	80.49	99.97	82.55	90.25	99.29	82.70	98.57	99.89	86.85	90.26	84.46
QMK (Q-kernel)	79.45	83.56	99.82	83.45	92.63	99.41	82.87	98.30	99.24	87.86	91.02	85.66

constraint set  $W$ . Therefore, an objective function can be constructed by using the ratio of  $d_W$  and  $d_B$  to get an optimization problem; the optimal  $A^*$  for the solution is

$$A^* = \arg \max_{A^T A = I} \frac{\text{tr}(A^T S_B A)}{\text{tr}(A^T S_W A)}. \quad (32)$$

Thus, calculating the Mahalanobis distance matrix is  $M^* = A^* (A^*)^T$  after learning.

**2.5. Procedural Steps and Discussion.** The procedure is shown in Figure 2. The procedure includes three stages: the multikernel optimization, training, and testing. The first step is optimizing the weight vector of multiple

TABLE 2: Performance on the Pavia University data (%).

Class	SVM (polynomial)	SVM (Gaussian)	MKL (polynomial)	MKL (Gaussian)	KSRC (polynomial)	KSRC (Gaussian)	Proposed (Q-kernel)
1	83.93	84.45	84.24	88.22	84.93	85.42	92.24
2	85.23	91.45	90.29	93.15	86.26	92.45	94.36
3	70.24	74.14	74.61	78.23	71.27	75.13	83.27
4	87.27	90.47	90.26	89.22	88.28	91.45	91.93
5	96.28	97.62	97.23	97.24	97.23	98.65	98.73
6	70.41	78.77	77.47	84.26	71.47	79.72	84.15
7	69.33	71.13	70.16	76.27	69.92	72.17	82.62
8	76.24	81.53	80.22	82.93	77.26	82.56	86.27
9	98.56	99.42	99.24	99.26	98.85	99.93	99.93

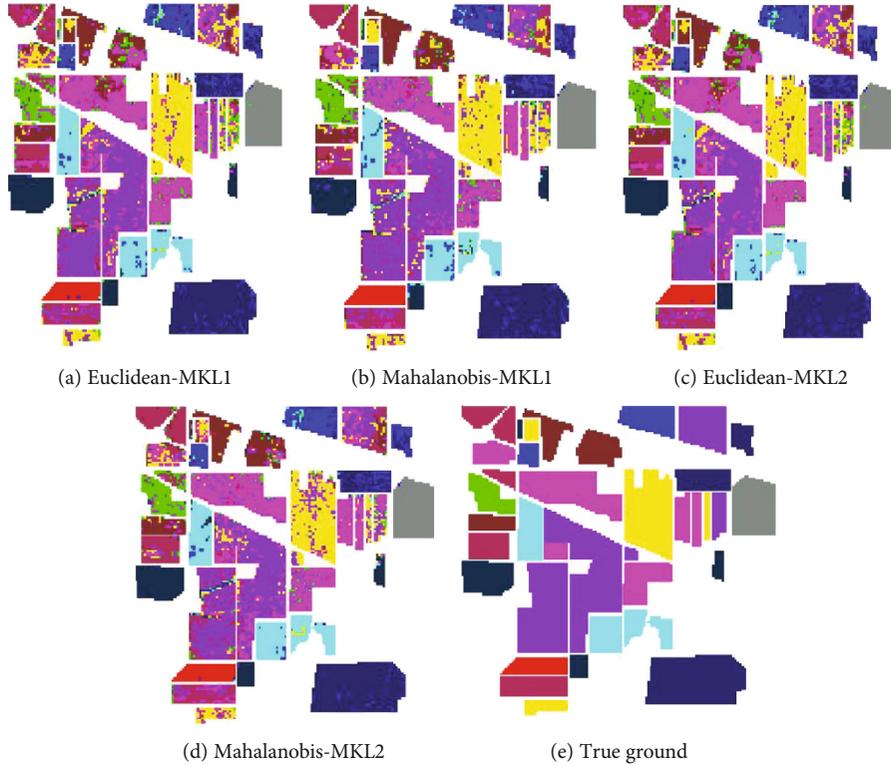


FIGURE 5: Classification results of different methods on the Indian Pines dataset.

quasiconformal kernels with Fisher and maximum margin criterions; the second step is optimizing multiple kernels for the classification. And the final step is classifier design with the optimal kernels.

On learning bounds of the proposed algorithm, the learning criterion is to maximize the accuracy of the test data. Given the function  $f : x \rightarrow R$ , the threshold version of  $f$  is defined as

$$er(f) = \frac{1}{n} |\{n+1 \leq i \leq 2n : y_i f(x_i) \leq 0\}|, \quad (33)$$

where the kernel-based classifiers are the threshold of kernel expansions of the form, and the bounded norm is

$$\|w\|^2 = \sum_{i,j=1}^{2n} \alpha_i \alpha_j k(x_i, x_j) = \alpha^T K \alpha \leq \frac{1}{\gamma^2}. \quad (34)$$

For any  $\gamma > 0$  with a probability at least  $1 - \delta$  over the data  $x_i, y_i$ , each function  $f \in F_k$  has  $er(f)$  no more than  $1/n \sum_{i=1}^n \max(1 - y_i f(x_i)) + 1/\sqrt{n}(4 + \sqrt{2 \log(1/\delta)}) +$

TABLE 3: OA (%) of different methods on the Indian Pines dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	57.23 ± 2.13	65.42 ± 1.95	67.98 ± 1.90	72.05 ± 1.24	74.23 ± 1.16
Mahalanobis-MKL1	62.25 ± 2.45	67.84 ± 2.33	71.97 ± 1.51	74.69 ± 1.23	75.10 ± 0.98
Euclidean-MKL2	59.10 ± 2.24	66.75 ± 1.82	71.08 ± 1.92	73.90 ± 1.24	76.07 ± 1.21
Mahalanobis-MKL2	62.04 ± 2.34	68.63 ± 2.43	73.34 ± 1.59	74.97 ± 1.29	77.12 ± 1.10

TABLE 4: Kappa coefficient of different methods on the Indian Pines dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	0.532	0.598	0.595	0.678	0.696
Euclidean-MKL2	0.544	0.612	0.663	0.698	0.728
Mahalanobis-MKL1	0.565	0.641	0.682	0.712	0.732
Mahalanobis-MKL2	0.570	0.648	0.689	0.729	0.742

$\sqrt{\zeta(k)/n\gamma^2}$ ), where  $\zeta(K) = E \max_{K \in K} \sigma^T K \sigma$  is the expectation over  $\sigma$ . So,  $\zeta(K) = cE \max_{K \in K} \sigma^T K / (\text{trace}(K)) \sigma$  and

$$\zeta(K) \leq c \min \left( m, n \max_j \frac{\lambda_j}{\text{trace}(K_j)} \right), \quad (35)$$

where  $\lambda_j$  is the largest eigenvalue of  $K_j$ .

### 3. Experiments and Analysis

**3.1. Experiment Setting.** The performance of the proposed intelligent hyperspectral instrument is evaluated. The accuracy of spectrum classification is an important index to evaluate the performance of spectrum classification. The experiment was carried out on two sensing datasets of a hyperspectral imager, i.e., Indian Pines dataset and Pavia University dataset. The dataset of Indian Pines is based on an airborne platform, under the various spectral and the spatial resolutions. The data includes 224 0.4-2.5  $\mu\text{m}$  bands. Nine kinds of 145  $\times$  145-pixel images are realized in the experiment. The data collection at the University of Pavia is based on a reflective optics system imaging spectrometer (ROSIS). The data includes 115 bands. In the experiment, the performance of 9 kinds of 610  $\times$  340 images is verified. Except for the feature dimensions of the participating categories, the rest of the two experiments were identical. In the use of classification features, considering the computational efficiency and stability of the Mahalanobis matrix, the dimension of original spectral features is reduced by PCA. After the dimension reduction, the features are normalized to eliminate the deviation caused by the sampling method. For the first experiment, the top 30 principal components are selected to participate in the classification; that is, the feature dimension was 30. For experiment 2, the first 40 principal components are

selected to participate in the classification; that is, the feature dimension is 40. On the classifier settings, the preset parameter values are selected by cross-validation by a standard multiclass SVM. In the kernel function setting, the Gaussian kernel function and the Mahalanobis Gaussian kernel function are used as the basis kernel functions, respectively. The scale parameter  $\sigma$  is set between [0.01, 2], and the number of basis kernels is 10. In terms of the evaluation index, the overall classification accuracy (OA) and Kappa coefficient (KC) are used as performance evaluation indicators, and information such as classifier training time, test time, and support vector number is collected. In the comparison method, the average multi-kernel and different multiple kernel learning methods are used as the multikernel combination coefficient algorithm, and the Euclidean distance Gaussian kernel and the Mahalanobis Gaussian kernel are, respectively, used for comparison. In these experiments, we implement four algorithms as follows.

The Indian Pines dataset is collected under various spectral and spatial resolutions. The spectral curves denote the different remote sensing environments with an airborne platform. The data cube has 224 bands of spectral resolution through 0.4-2.5  $\mu\text{m}$  range, and it has the spatial resolution of 20 m per pixel. We removed the noisy and water vapor absorption bands, and 200 bands of images are used in the experiments. The whole scene consists of 145  $\times$  145 pixels and 16 classes of interested objects with the size ranging from 20 to 2468 pixels; 9 classes are used in the experiments. One example is shown in Figure 3.

Pavia University data was acquired by the reflective optics system imaging spectrometer (ROSIS) over the urban area of the University of Pavia, Northern Italy. The dataset consists of 115 spectral bands and 610  $\times$  610 pixels with the spatial resolution of 1.3 m by pixel. Several undesirable bands influenced by the atmospheric absorption are discarded, leaving 103 bands in the 0.43-0.86  $\mu\text{m}$  region. We cut a patch sized 610  $\times$  340, consisting of 9 classes of land covers from the set. The example is shown in Figure 4.

**3.2. Experiments on the Performance on Quasiconformal Kernel Mapping.** In the experiments, we evaluate the performance on quasiconformal kernel mapping on the two databases. The performance of quasiconformal kernel mapping is testified and evaluated with the polynomial kernel and Gaussian kernel. We have the Kernel Sparse Representation Classifier (KSRC) and Support Vector Classifier (SVC) for

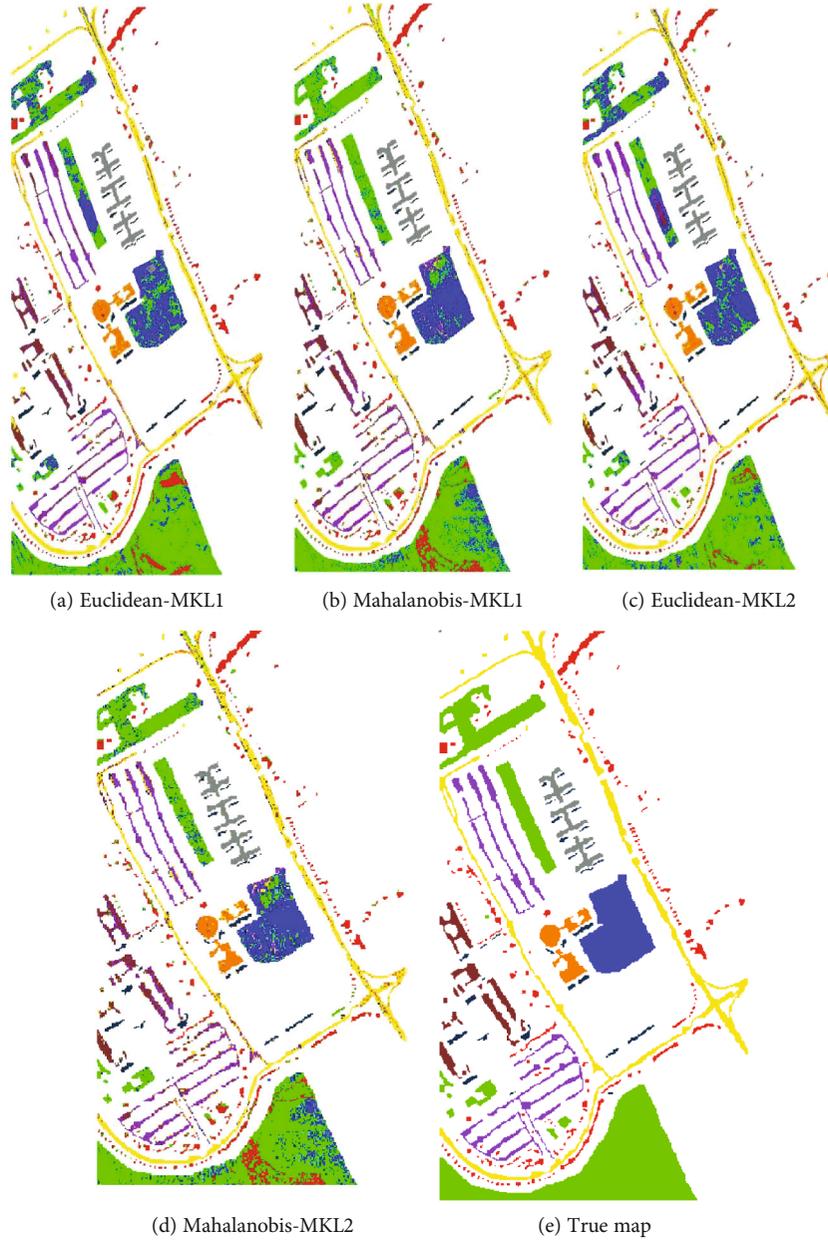


FIGURE 6: Classification results of different methods on the Pavia University dataset.

TABLE 5: OA (%) of different methods on the Pavia University dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	$61.98 \pm 3.16$	$67.78 \pm 3.24$	$71.35 \pm 2.39$	$75.35 \pm 1.62$	$78.69 \pm 1.12$
Mahalanobis-MKL1	$62.53 \pm 3.23$	$69.34 \pm 2.78$	$72.80 \pm 2.10$	$76.24 \pm 1.19$	$79.78 \pm 1.05$
Euclidean-MKL2	$61.92 \pm 3.21$	$66.97 \pm 3.28$	$71.76 \pm 2.39$	$75.56 \pm 1.67$	$79.24 \pm 1.33$
Mahalanobis-MKL2	$62.89 \pm 3.22$	$69.42 \pm 2.81$	$73.10 \pm 2.23$	$76.34 \pm 1.23$	$79.86 \pm 1.16$

classification. For comparisons, we also implement other algorithms, including SVM [29], RMKL-SVM [30], and POL-KSRC [31]. The experimental results on two datasets are shown in Tables 1 and 2.

3.3. *Experiments on the Performance on Mahalanobis Distance Kernel.* In the comparison method, the average multikernel and different multiple kernel learning methods are used as the multikernel combination coefficient algorithm,

TABLE 6: Kappa coefficient of different methods on the Pavia University dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	0.548	0.608	0.658	0.704	0.721
Euclidean-MKL2	0.549	0.611	0.662	0.710	0.729
Mahalanobis-MKL1	0.554	0.619	0.672	0.716	0.734
Mahalanobis-MKL2	0.555	0.622	0.678	0.721	0.741

TABLE 7: Performance comparisons on two databases.

Methods	Indian Pines dataset		Pavia University dataset	
	OA (%)	KC (%)	OA (%)	KC (%)
RBF	73.23	63.73	75.43	68.76
Poly	75.77	66.28	78.08	72.07
Mahal-RBF	76.92	67.79	76.26	69.87
Mahal-Poly	77.64	68.62	79.07	73.24
SK-CV-RBF	67.03	64.23	75.71	69.14
SK-Poly	69.37	66.96	77.62	71.37
NMF-MKL	67.48	64.81	71.57	64.42
KNMF-MKL	68.22	65.63	72.80	65.81
Euclidean-MKL1	74.23	69.63	78.69	72.11
Euclidean-MKL2	76.07	72.84	79.24	72.92
Mahalanobis-MKL1	75.90	73.25	79.78	73.43
Mahalanobis-MKL2	78.22	74.25	79.86	74.16
Proposed Mahalanobis-QMKL1	76.96	74.35	80.27	74.52
Proposed Mahalanobis-QMKL2	79.28	75.85	80.96	75.34

and the Euclidean distance Gaussian kernel and the Mahalanobis Gaussian kernel are, respectively, used for comparison. In these experiments, we implement four algorithms as follows: Euclidean-MKL1 [32]: the Euclidean distance kernel function. Each kernel function is combined according to the same weight; that is, the combination coefficient of each kernel function is the reciprocal of the number of kernel functions (see the description of literature [32] for details). Mahalanobis-MKL1: the Mahalanobis distance kernel function is used for kernel learning. Euclidean-MKL2 [33]: the Euclidean distance kernel function is used, which describes the combination coefficient as described in [33]. Mahalanobis-MKL2: the Mahalanobis distance kernel function is used, and the kernel learning is the same as Euclidean-MKL2.

The experimental results of different methods on the Indian Pines dataset are shown in Figure 5 and Tables 3 and 4. The experimental results of different methods on the Pavia University dataset are shown in Figure 6 and Tables 5 and 6. As these results, the proposed Mahalanobis distance kernel has the highest performance on the accuracy.

**3.4. Experiment Comparisons.** For the comparisons, we have some experiments to compare the performance of the proposed algorithm, and the following 14 methods are imple-

mented as the comparison: (1) RBF: RBF Euclidean kernel as the kernel function in the SVM [31]; (2) Poly: polynomial Euclidean kernel as the kernel function in the SVM [31]; (3) Mahal-RBF: Mahalanobis distance-based RBF kernel as the kernel function in the SVM [34]; (4) Mahal-Poly: Mahalanobis distance-based polynomial kernel as the kernel function in the SVM [34]; (5) SK-CV (RBF): a SVM with a single kernel and adopting the RBF kernel as the kernel function in the SVM [35]; (6) SK-Poly: standard SVM with a single kernel and adopting a polynomial kernel as the kernel function in the SVM [35]; (7) NMF-MKL: the nonnegative matrix factorization (NMF) MKL proposed by Gu et al. [28], which combines multiple kernels with NMF; (8) KNMF-MKL: the kernel-based nonnegative matrix factorization (KNMF) MKL method, also proposed by Gu et al., which combines multiple kernels with the KNMF method; (9) Euclidean-MKL1 [32]: the Euclidean distance kernel function. Each kernel function is combined according to the same weight; that is, the combination coefficient of each kernel function is the reciprocal of the number of kernel functions (see the description of literature for details). (10) Euclidean-MKL2 [33]: the Euclidean distance kernel function is used, which describes the combination coefficient as described in [33]; (11) Mahalanobis-MKL1: proposed the Mahalanobis distance-based multiple kernel function, and learning criterions are same as Euclidean-MKL1; (12) Mahalanobis-MKL2: the Mahalanobis distance kernel function is used, and learning criterions are the same as Euclidean-MKL2; (13) Mahalanobis-QMKL1: proposed the Mahalanobis distance-based multiple quasiconformal kernel function, and learning criterions are the same as Euclidean-MKL1; (14) Mahalanobis-QMKL2: the Mahalanobis distance quasiconformal kernel function is used, and learning criterions are the same as Euclidean-MKL2.

As shown in Table 7, the proposed scheme is effective to the hyperspectral image classification. The quasiconformal mapping-based multiple kernel learning network framework is effective and feasible for hyperspectral data classification, and the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Compared with other kernel-based learning methods, the proposed algorithm is effective and performs best.

**3.5. Computation Efficiency and Practical Applications.** In the experiments, the computational cost was recorded with a PC with a 2.6 GHz i5-3320 processor and 4 GB RAM. The different computational costs are achieved under the different features. The proposed algorithms omit the parameter optimization process under the same dimension of feature vector, so the high computation efficiency is achieved. Both Euclidean-based method and Mahalanobis-based method adopt nonnegative matrix factorization to optimize the kernel weights, and a higher dimension of features requires more time because it needs more memory to save the kernel matrix and has more dimensions to compute. So, Mahalanobis-based kernel learning has the higher

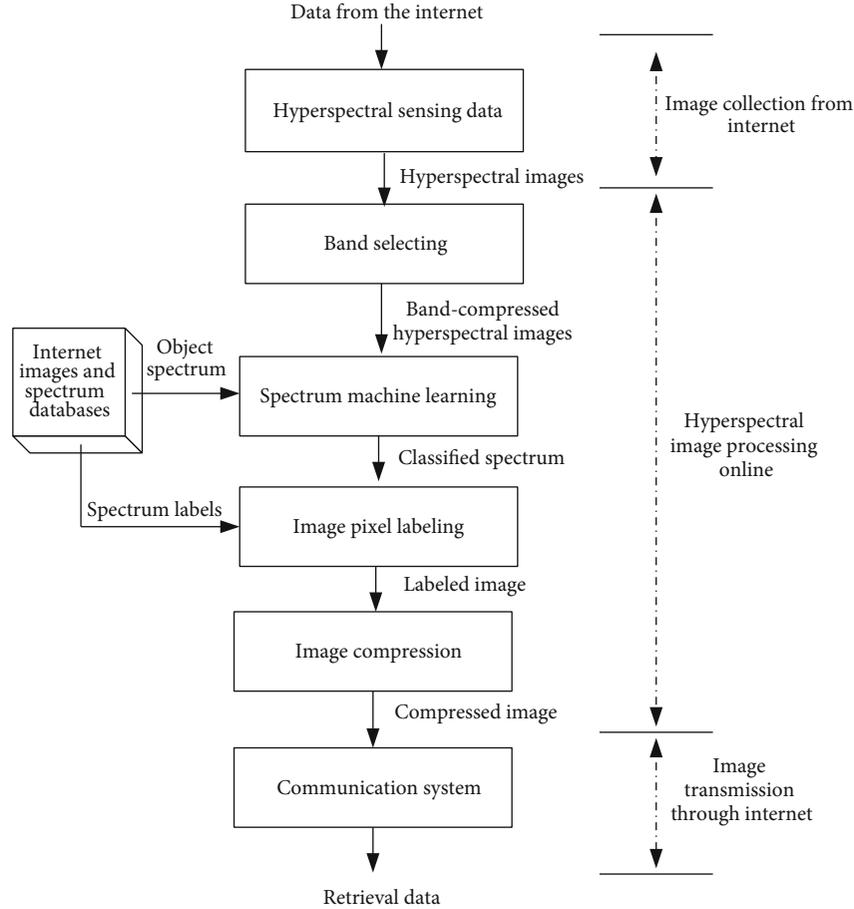


FIGURE 7: The practical application framework of hyperspectral data retrieval under internet environment.

computation efficiency than Euclidean-based method under the same dimension of the features.

In the practical application system, the system framework is shown in Figure 7. For the quasiconformal mapping kernel machine learning-based intelligent hyperspectral data retrieval under internet environment, the framework includes three stages: image collection, image processing online, and image transmission. In the framework, the second stage is important. Different from the experiment setting, the practical application system includes the image collection and image transmission.

#### 4. Conclusion

In this paper, we present the quasiconformal mapping kernel machine learning-based intelligent hyperspectral data classification algorithm for the internet-based hyperspectral data retrieval and with the wide application in intelligent internet data mining as the application of AIoT (Artificial Intelligence of Things). The contributions of the algorithm lie in the following points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification, and the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring

the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Experiments show that the proposed scheme is effective to the hyperspectral image classification. The proposed algorithm has advantages on the large training sample construction with the data from the internet, including images, videos, and other information.

#### Data Availability

We have not used specific data from other sources for the simulations of the results. The two popular hyperspectral datasets in this paper, Indian Pines dataset and Pavia University data, are downloaded free from the website: [http://www.ehu.es/ccwintco/index.php?title=Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes). The proposed algorithm is implemented in Python.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Acknowledgments

We would like to thank Dr. Li Li, Prof. Junbao Li, and Prof. Yanfeng Gu for providing the kernel-based learning

programs of their papers for comparison in the experiments. This work is supported by the National Science Foundation of China under Grant No. 61871142.

## References

- [1] A. L. Afzal and S. Asharaf, "Deep kernel learning in core vector machines," *Pattern Analysis & Applications*, vol. 21, no. 3, pp. 721–729, 2018.
- [2] J.-B. Li, J.-S. Pan, and S.-C. Chu, "Kernel class-wise locality preserving projection," *Information Sciences*, vol. 178, no. 7, pp. 1825–1835, 2008.
- [3] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.
- [4] L. Wang, L. Chan, and P. Xue, "A Criterion for Optimizing Kernel Parameters in KBDA for Image Retrieval," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 3, pp. 556–562, 2005.
- [5] J. S. Pan, P. Hu, and S. C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *PRO*, vol. 7, no. 11, pp. 845–856, 2019.
- [6] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on Toeplitz matrix-vector product approach," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 27, no. 7, pp. 1614–1622, 2019.
- [7] H. Zhao, S. Sun, Z. Jing, and J. Yang, "Local structure based supervised feature extraction," *Pattern Recognition*, vol. 39, no. 8, pp. 1546–1550, 2006.
- [8] J. Huang, P. Yuen, W. Chen, and J. Lai, "Kernel subspace LDA with optimized kernel parameters on face recognition," *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004, pp. 187–192, Seoul, South Korea, South, May 2004.
- [9] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.
- [10] X. Xie, B. Li, and X. Chai, "A framework of quasiconformal mapping-based kernel machine with its application to hyperspectral remote sensing," *Measurement*, vol. 80, pp. 270–280, 2016.
- [11] N. Subrahmanya and Y. Shin, "Sparse multiple kernel learning for signal processing applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 788–798, 2010.
- [12] S. Sören, G. Rätsch, and C. Schäfer, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.
- [13] K. Marius, U. Brefeld, and S. Sonnenburg, "lp-norm multiple kernel learning," *The Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.
- [14] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.
- [15] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen, "Scalable high-performance image registration framework by unsupervised deep feature representations learning," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1505–1516, 2016.
- [16] M. Jiu and H. Sahbi, "Nonlinear deep kernel learning for image annotation," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1820–1832, 2017.
- [17] C. Wang, J. Shi, Q. Zhang, and S. Ying, "Histopathological image classification with bilinear convolutional neural networks," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4050–4053, Seogwipo, South Korea, 2017.
- [18] W. Li, X. Qian, and J. Ji, "Noise-tolerant deep learning for histopathological image segmentation," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3075–3079, Beijing, China, September 2017.
- [19] N. Kumar, A. V. Rajwade, S. Chandran, and S. P. Awate, "Kernel generalized Gaussian and robust statistical learning for abnormality detection in medical images," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 4157–4161, Beijing, China, September 2017.
- [20] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep multilayer multiple kernel learning," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2305–2314, 2016.
- [21] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep kernel-SVM network," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1955–1960, Vancouver, BC, Canada, July 2016.
- [22] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.
- [23] E. K. Wang, C. M. Chen, F. Wang, M. K. Khan, and S. Kumari, "Joint-learning segmentation in Internet of drones (IoD)-based monitor systems," *Computer Communications*, vol. 152, pp. 54–62, 2020.
- [24] J. Zhuang, I. W. Tsang, and S. C. H. Hoi, "Two-layer multiple kernel learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 909–917, Ft. Lauderdale, 2011.
- [25] E. V. Strobl and S. Visweswaran, "Deep multiple kernel learning," in *2013 12th International Conference on Machine Learning and Applications*, no. 1, pp. 414–417, Miami, FL, USA, December 2013.
- [26] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep architecture using multi-kernel learning and multi-classifier methods," in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–6, Marrakech, Morocco, November 2015.
- [27] L. Le, J. Hao, Y. Xie, and J. Priestley, "Deep kernel: learning kernel function from data using deep neural network," in *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*, pp. 1–7, Shanghai, China, December 2016.
- [28] Y. Gu, Q. Wang, H. Wang, D. You, and Y. Zhang, "Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2739–2751, 2015.
- [29] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "Simple MKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
- [30] S. Zhuo, W. Cheng, and H. Ying, "Multiple kernel learning for classification of hyperspectral imagery with neighborhood preserving embedding," *The Journal of Computer Information Systems*, vol. 9, no. 9, pp. 3563–3570, 2013.

- [31] D. Wang, D. S. Yeung, and E. C. C. Tsang, "Weighted Mahalanobis distance kernels for support vector machines," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1453–1462, 2007.
- [32] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
- [33] Y. Gu, C. Wang, Y. Z. Di You, Y. Zhang, S. Wang, and Y. Zhang, "Representative multiple kernel learning for classification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 7, pp. 2852–2865, 2012.
- [34] L. Li, C. Sun, L. Lin, J. Li, and S. Jiang, "A Mahalanobis metric learning-based polynomial kernel for classification of hyperspectral images," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1103–1113.
- [35] L. Li, C. Sun, L. Lin, J. Li, and S. Jiang, "A dual-layer supervised Mahalanobis kernel for the classification of hyperspectral images," *Neurocomputing*, vol. 214, pp. 430–444, 2016.