Hindawi
Complexity
Volume 2019, Article ID 1806314, 18 pages
https://doi.org/10.1155/2019/1806314

WILEY | Hindawi

Research Article

Globality-Locality Preserving Maximum Variance Extreme Learning Machine

Yonghe Chu,[1] **Hongfei Lin**,[1] **Liang Yang**,[1] **Yufeng Diao**,[1] **Dongyu Zhang**,[1] **Shaowu Zhang**,[1] **Xiaochao Fan**,[1] **Chen Shen**,[1] **and Deqin Yan**[2]

[1]*Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China*
[2]*School of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China*

Correspondence should be addressed to Hongfei Lin; hflin@dlut.edu.cn

Received 5 December 2018; Revised 26 February 2019; Accepted 1 April 2019; Published 2 May 2019

Academic Editor: Michele Scarpiniti

An extreme learning machine (ELM) is a useful technique for machine learning; however, the existing extreme learning machine methods cannot exploit the geometric structure information or discriminate information of the data space well. Therefore, we propose a globality-locality preserving maximum variance extreme learning machine (GLELM) based on manifold learning. Based on the characteristics of the traditional ELM method, GLELM introduces the basic principles of linear discriminant analysis (LDA) and local preservation projection (LPP) into ELM, fully taking account of the discriminant information contained in the sample. This method can preserve the global and local manifold structures of data to optimize the projection direction of the classifier. Experiments on several widely used image databases and UCI datasets validate the performance of GLELM. The experimental results show that the proposed model achieves promising results compared to several state-of-the-art ELM algorithms.

1. Introduction

Single-layer feedforward networks (SLFNs) have been intensively studied over the past several decades. The well-known algorithm in single layer feedforward networks is the backpropagation (BP) algorithm proposed by Rumelhar et al. [1] in 1986. The BP algorithm uses the idea of gradient descent to optimize the parameters in the neural network, but this optimization has the disadvantages of slow training speed, and it easily falls into a local minimum. Therefore, researchers have proposed different improved algorithms for the problem of slow training speed which easily falls into a local minimum. Hagan et al. [2] proposed a second-order optimization method in 1994. Branke et al. [3] proposed a global optimization method in 1995. Li et al. [4] proposed a subset selection method in 2005.

Recently, the extreme learning machine (ELM) [5] has attracted increasing attention from scholars. ELM is developed on the basis of single-hidden layer feedforward networks (SLFNs) and can be regarded as an extension of SLFNs. In traditional neural network algorithms, for example, the BP[1] neural network uses the gradient descent-based method to adjust the input weight and basis value of the hidden layer nodes in an iterative manner. However, the method based on gradient descent has the disadvantages of slow solution speed and easily falling into a locally optimal solution. Compared with the traditional neural network algorithm, ELM randomly generates the input weight and basis value of the hidden layer node, so it has a faster solution speed and requires less human intervention during the training process. The literature [6, 7] analysed the input weight and bias value of ELM at randomly generated hidden layer nodes to determine the output weight, which maintained the general approximation ability of SLFNs. At the same time, a near-global optimal solution can be obtained. The literature [8, 9] notes that ELM has better classification performance than support vector machine (SVM) [10]. Due to the good generalization ability of ELM, ELM has been widely used in pattern recognition [11–15].

In recent years, researchers have studied ELM in various ways and proposed various improvements. Huang et al. [6] further studied the general approximation ability of ELM. Lin

et al. and Liu et al. [16–18] used statistical learning theory to conduct in-depth research on the generalization ability of ELM. Wang [19] et al. proposed a local generalization error model for the problem of ELM generalization ability, and the researchers also compared ELM with other classification algorithms. Shi et al. [20] studied ELM and SVM and their improved algorithms in depth and concluded that ELM is superior to SVM in training speed and generalization ability. Many variants of ELM have been proposed to meet particular application requirements. For example, Wang et al. [21] analysed the influence of the hidden layer node output matrix on the ELM algorithm and proposed an improved algorithm. Zheng et al., Riccardo et al., and Zhang et al. [22–24] proposed various improvements to the ELM algorithm by analysing the influence of data on the ELM model from the perspective of cost sensitivity coefficients. Li [25] et al. studied the defects of ELM in unbalanced data and missing data to improve the ELM algorithm. Zhou et al. and Javier [26–28] et al. applied ELM to remote sensing images. Zhou et al. [29] proposed various improvements for ELM to solve the problems in online continuous data applications. Recently, researchers have combined ELM and dimensionality reduction techniques for application. Castaño [30] et al. applied principal component analysis (PCA) dimensionality reduction techniques to ELM, and Wang et al. [31] combined the local tangent space alignment (LTSA) dimensionality reduction algorithm with ELM. Researchers have also applied integration techniques to ELM to improve the robustness of ELM algorithms. Zhang et al. [32] applied AdaBoost technology to ELM, and Liu et al. [33] proposed an integrated extreme learning machine. Deepak et al. [34] applied bagging technology to the ELM algorithm.

The above improvements in theory and application enhance the generalization capability of ELM and greatly expand the application range of the ELM algorithm. However, the discriminant information of the ELM algorithm on the data samples and the global and local manifold structures between the data samples have not yet been carefully studied in mathematics or geometry. Recently, researchers have noted that manifold learning methods [35, 36] can effectively reveal the intrinsic geometry of data points [9]. Assuming that data samples x_1 and x_2 are drawn from the same marginal distribution $P_{\boldsymbol{x}}$, if two points x_1 and x_2 are close to each other, then the conditional probabilities $P(\boldsymbol{y} \mid \boldsymbol{x}_1)$ and $P(\boldsymbol{y} \mid \boldsymbol{x}_2)$ should be similar as well. The above assumptions are widely referred to as smoothing assumptions in machine learning. Therefore, by mining the geometry between the data, it is possible to provide effective information for pattern classification. Recently, the researchers carried on the thorough research on manifold learning, puts forward the different methods to keep local characteristics of data [37–39]. Aiming to solve the drawback of ELM that the intrinsic manifold structure of the data space is ignored, and inspired by manifold learning and literature [40], we introduce the basic principles of linear discriminant analysis (LDA) [41] and locality preserving projections (LPP) [42] into ELM, proposing a novel learning algorithm called the globality-locality preserving maximum extreme learning machine (GLELM) in which the manifold structure within each class is explicitly considered. This method introduces the intraclass divergence and interclass divergence matrix in LDA and the basic principle of LPP into ELM so that it not only maintains the intrinsic local geometry of the sample but also maintains the global geometric structure of the sample to a certain extent and embodies the global discriminant information contained in the sample. GLELM retains the locality preserving characteristic of LPP and utilizes the global discriminative structures obtained from MMC, which can maximize the between-class distance and minimize the within-class distance. We combine the thought of LPP and the principle of LDA into ELM model, to enhance the information discriminant ability of ELM. So GLELM is superior to ELM for recognition task. Moreover, the experimental results show that the intrinsic manifold structure of the data sample can effectively improve the classification performance of the ELM algorithm. In addition, the literature [43] noted that some recent research shows that the images will reside on a nonlinear submanifold. Therefore, in this case, GLELM can usually achieve better performance than ELM. The contributions of the GLELM algorithm mentioned in this paper are as follows.

(1) While inheriting the characteristics of ELM, GLELM avoids the problem of insufficient learning to some extent.

(2) The basic principles of LDA and LPP are introduced into ELM, which effectively maintains the intrinsic local geometry and global geometry of the sample and introduces the global discriminant information of the data samples into the ELM model.

(3) The idea of manifold learning is applied to the ELM model, and the validity of the GLELM algorithm is verified by experiments.

The rest of the paper is organized as follows. In Section 2, this paper introduces related work. In Section 3, we introduce the basic principles and framework of the ELM algorithm. Section 4 presents the GLELM algorithm framework. Section 5 describes and analyses the experimental results. Section 6 summarizes the paper.

2. Background and Notation

2.1. Notations. Given datasets $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in R^{D\times N}$, D is data dimension, $N = N_1 + N_2 + \cdots + N_C$ is the number of samples, C is the total number of categories for datasets. The dataset label vector is $[y_1, y_2, \ldots, y_N] \in R^N$. Define the projection transformation matrix as $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d] \in R^{D\times d}$. The data are reduced from the original space R^D to the low-dimensional subspace R^d. The symbols $\|\cdot\|_2$ express the l_2-norm. $Tr(\cdot)$ denotes the trace operator; $N_k(\cdot)$ denotes the k nearest neighbours operator.

2.2. Related Works. In this section, we briefly review the related work. Iosifidis et al. used the principle of linear discriminant analysis to explore the geometric structure of the data and introduced the intraclass divergence matrix and the global divergence matrix into the ELM model and proposed a minimum class variance extreme learning machine (MCVELM) [44] and a minimum variance extreme learning machine (MVELM) [45], respectively. On this basis, Iosifidis

et al. proposed a graph embedding extreme learning machine (GEELM) [46] to optimize the network output weights of ELM. The GEELM provides a unified way to incorporate subspace learning criteria formulated using graphs in elm optimization. In their paper, formulations using supervised and unsupervised subspace criteria in elm optimization are used. Liu et al. proposed the robust discriminative extreme learning machine (RDELM) [47] for the deficiency of the MCVELM algorithm for discriminating information between data samples. The RDELM algorithm not only takes into account the intraclass discrimination information of the data samples but also considers the interclass discrimination information of the data samples. The motivation for our paper is similar to the above papers, which also discussed the geometry of ELM. However, they directly used the geometric structure information of the data to optimize the network output weight. We focus on the data samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ that are drawn from the same marginal distribution $P_{\boldsymbol{x}}$. If two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are close to each other, then the conditional probabilities $P(\boldsymbol{y} \mid \boldsymbol{x}_1)$ and $P(\boldsymbol{y} \mid \boldsymbol{x}_2)$ should be similar as well; therefore, the manifold structure information of the data samples is introduced into the ELM model, and the generalization ability of the ELM algorithm is enhanced.

The most relevant work was proposed by literature [48–50]. Iosifidis et al. introduced local class information into the ELM model and proposed a Local Class Variance Extreme Learning Machine (LCVELM) classifier [48]. Based on the consistency property of data, which enforces similar samples to share similar properties, Peng et al. proposed a discriminative graph regularized extreme learning machine (GELM) [49]. GELM constructs the Laplacian Eigenmap (LE) [51] structure with discriminant information of data samples and introduces it into the ELM algorithm as a regular term. In addition, Peng et al. proposed a discriminative manifold extreme learning machine (DMELM) [50] based on local intraclass discriminant information, local interclass discriminant information, and data geometric structure information. The GELM and DMELM algorithms proposed by Peng et al. enhance the classification performance and generalization ability of the ELM model by introducing the manifold structure and discriminant information of the data samples into the ELM model. However, the GELM and DMELM algorithms ignore the global geometry and discriminant information of the data samples. The literature [52] shows that the intraclass divergence matrix, interclass divergence matrix, and global divergence matrix in linear discriminant analysis (LDA) maintain the global discriminant information and global geometric structure of the training samples. Therefore, based on the basic principles of the LDA and LPP algorithms, we introduce the global and local manifold structure and discriminant information into the ELM model and propose the GLELM model.

It is clear that our GLELM models are the natural extension of ELM with the manifold regularization, and the manifold learning methods have also been combined with other machine learning algorithms, such as globality–locality preserving projections (GLPP) [53, 54] and support vector machine with globality-locality preserving (GLPSVM) [55]; GLPP separates the data into a static part (subject-invariant factors) and a dynamic part (intrasubject factors) at first and then jointly learns these two graph Laplacians to yield a new graph Laplacian. GLPP realize dimensionality reduction for data by using the aforementioned method. By using LPP to keep local geometry information and LDA to keep global geometry information of data, GLELM unifies LPP and LDA into a manifold regularization framework. The proposed GLELM algorithm combines manifold criterion and Fisher criterion, with a stronger discriminative ability. GLPSVM introduced manifold structure information into SVM, using geometry and discriminative information to construct manifold regularization framework. Both GLPSVM and GLELM use LPP to construct manifold framework; however, GLPSVM uses data sample mean vector to obtain the global geometric structure information of data while GLELM uses LDA. In addition, the architecture of GLELM is completely different from the GLPP and GLPSVM. In fact, GLPP is a dimensionality reduction algorithm. As a classification algorithm GLPSVM do the classification by maximizing the geometric intervals. Based on single hidden layer feedforward neural network, GLELM randomly generate output weights and hidden layer offset value and analyse and determine the weights of the output so as to realize the data classification. Different architecture leads to different recognition performance.

2.3. Extreme Learning Machine. The extreme learning machine proposed by Huang et al. [5] is an efficient and practical learning mechanism for single-layer feedforward neural networks. For N different samples $K = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) \mid \boldsymbol{x}_i \in R^d, \boldsymbol{t}_i \in R^m, i = 1, 2, \ldots, N\}$, where $\boldsymbol{x}_i = (x_1, x_2, \ldots, x_{id})^T$ and $\boldsymbol{t}_i = (t_1, t_2, \ldots, t_{im})^T$, the ELM model with L hidden layer node activation function $g(\boldsymbol{x})$ is as follows:

$$\sum_{j=1}^{L} \boldsymbol{\beta}_j g_j(\boldsymbol{x}_i) = \sum_{j=1}^{L} \boldsymbol{\beta}_j g\left(\boldsymbol{a}_j \cdot \boldsymbol{x}_i + b_j\right) = \boldsymbol{o}_i \tag{1}$$

where $\boldsymbol{a}_j = (a_{i1}, a_{i2}, \ldots, a_{id})$ is the input weight vector connecting the jth hidden layer node with the input nodes; $\boldsymbol{\beta}_j = (\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm})^T$ is the output weight vector connecting the jth hidden layer node and the output node. b_j is the offset value of the jth hidden layer node. $\boldsymbol{a}_j \cdot \boldsymbol{x}_i$ represents the inner product of $\boldsymbol{a}_j$ and $\boldsymbol{x}_i$. $\boldsymbol{o}_i = (o_{i1}, o_{i2}, \ldots, o_{im})^T$ is the network output corresponding to sample $\boldsymbol{x}_i$. To integrate all data samples, (1) can be rewritten as follows:

$$\boldsymbol{\beta}^T \boldsymbol{H} = \boldsymbol{T} \tag{2}$$

where $h(\boldsymbol{x}_i) = (g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_i + b_1), g(\boldsymbol{a}_2 \cdot \boldsymbol{x}_i + b_2), \ldots, g(\boldsymbol{a}_L \cdot \boldsymbol{x}_i + b_L))^T$ is the output vector of the hidden layer with respect to $\boldsymbol{x}_i$, $\boldsymbol{H}$ is the network hidden layer node output, $\boldsymbol{\beta}$ is the output weight matrix, and $\boldsymbol{T}$ is the expected output matrix:

$$\begin{aligned} \boldsymbol{H} &= \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_N + b_1) \\ \vdots & \vdots & \vdots \\ g(\boldsymbol{a}_L \cdot \boldsymbol{x}_1 + b_L) & \cdots & g(\boldsymbol{a}_L \cdot \boldsymbol{x}_N + b_L) \end{bmatrix}_{L \times N} \\ &= [h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_N)] \end{aligned} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L\times m}, \quad T = [\boldsymbol{t}_1, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_1]_{m\times N} \tag{4}$$

The standard single hidden feedforward neural networks (SLFNs) are to compute appropriate $\widetilde{\boldsymbol{a}}_j$, $\widetilde{\boldsymbol{b}}_j$, and $\widetilde{\boldsymbol{\beta}}$ ($j = 1, 2, \ldots, L$) to satisfy

$$\begin{aligned} &\left\| H\left(\widetilde{\boldsymbol{a}}_1, \widetilde{\boldsymbol{a}}_2, \ldots, \widetilde{\boldsymbol{a}}_L, \widetilde{\boldsymbol{b}}_1, \widetilde{\boldsymbol{b}}_2, \ldots, \widetilde{\boldsymbol{b}}_L\right) - \boldsymbol{T}\right\|^2 \\ &\quad = \min_{\boldsymbol{a}_j, \boldsymbol{b}_j, \boldsymbol{\beta}} \left\| H\left(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_L, \boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_L\right) - \boldsymbol{T}\right\|^2 \end{aligned} \tag{5}$$

Formula (5) can be solved by gradient descent method. Huang et al. [22] have proved that the weights between input layer and the biases need no adjustment compared with the standard SLFNs. In the algorithm of ELM, weights and bias values of hidden layer nodes are randomly input; the single hidden layer feedforward neural network nonlinear model is converted into linear model. Formula (5) can be written as $\boldsymbol{\beta}^T\boldsymbol{H} = \boldsymbol{T}$ and can be solved by using least square method. When the number of hidden layer nodes is the same as the number of training samples ($L = N$), we can directly obtain the optimal output weight matrix $\boldsymbol{\beta}$ by the inverse matrix of matrix $\boldsymbol{H}$ by (5). However, in most cases, the number of hidden layer nodes is much smaller than the number of training samples ($L < N$). At this time, the matrix $\boldsymbol{H}$ is a singular matrix. We solve (5) by the least squares solution:

$$\widetilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{H} - \boldsymbol{T}\right\|^2 = \boldsymbol{H}^{+}\boldsymbol{T} \tag{6}$$

where $\boldsymbol{H}^{+}$ is the generalized inverse matrix of the matrix and $\boldsymbol{H}^{+}$ can be calculated by SVD or least-squares.

To improve the stability and generalization capability of traditional ELM, Huang [22] proposed the equality optimization constraint-based ELM. The optimization formula of the ELM of the equality optimization constraint not only minimizes the training error $\boldsymbol{\varepsilon}$ but also minimizes the output weight $\boldsymbol{\beta}$, so the ELM target of the equality optimization constraint can be written as

$$\begin{aligned} \min_{W} \quad & \frac{1}{2}\left\|\boldsymbol{\beta}\right\|^2 + \frac{1}{2}C\sum_{i=1}^{N}\left\|\boldsymbol{\varepsilon}_i\right\|^2 \\ s.t. \quad & \boldsymbol{\beta}^{\mathrm{T}} h\left(\boldsymbol{x}_i\right) = \boldsymbol{t}_i - \boldsymbol{\varepsilon}_i \\ & i = 1, 2, \cdots, N \end{aligned} \tag{7}$$

In (7), $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \cdots, \varepsilon_{im})^T$ is a training error vector corresponding to the sample $\boldsymbol{x}_i$, and C is a penalty parameter.

The number of training samples is larger than the number of hidden layer nodes, or the number of training samples is smaller than the number of hidden layer nodes in the calculation process of ELM. The two cases corresponding to the output weight $\boldsymbol{\beta}$ are different. We will rewrite them as follows:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\left\|\boldsymbol{\beta}\right\|^2 + \frac{1}{2}C\left\|\boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{H} - \boldsymbol{T}\right\|^2 \tag{8}$$

When the number of training samples is less than the number of hidden layer nodes ($L > N$), the solution to (8) is

$$\boldsymbol{\beta} = \boldsymbol{H}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{T}^T \tag{9}$$

When the number of training samples is greater than the number of hidden layer nodes ($L < N$), the solution to (10) is

$$\boldsymbol{\beta} = \left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{H}\boldsymbol{T}^T \tag{10}$$

The ELM algorithm solving process can be summarized as follows:

(1) Initialize the training sample set

(2) Randomly specify the network input weight $\boldsymbol{a}_j$ and the offset value b_j, $j = 1, 2, \cdots, L$

(3) Calculate the hidden layer node output matrix $\boldsymbol{H}$ by the activation function

(4) Calculate the output weight matrix $\boldsymbol{\beta}$ according to (9) or (10)

2.4. Linear Discriminant Analysis. The main idea of LDA is to enhance the global class discrimination after projection, which maximizes the rank of the inter-class discrete matrix $\boldsymbol{S}_B$ by minimizing the rank of the intraclass discrete matrix S_W to find a subspace to distinguish different categories. According to the derivation of LDA in the literature [48], $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ are defined as follows:

$$\boldsymbol{S}_W = \sum_{i=1}^{C}\sum_{j=1}^{N_i}\left(\boldsymbol{h}_j^{(i)} - \boldsymbol{m}^{(i)}\right)\left(\boldsymbol{h}_j^{(i)} - \boldsymbol{m}^{(i)}\right)^T \tag{11}$$

$$\boldsymbol{S}_B = \sum_{i=1}^{C} N_i\left(\boldsymbol{m}^{(i)} - \boldsymbol{m}\right)\left(\boldsymbol{m}^{(i)} - \boldsymbol{m}\right)^T \tag{12}$$

In (11) and (12), N_i is the number of samples in the ith class, and $\boldsymbol{h}_j^{(i)}$ is the jth sample in the ith class. $\boldsymbol{m}^{(i)}$ is the mean vector of the ith class, $\boldsymbol{m}$ represents the mean vector of all samples, and C is the total number of categories in the dataset. LDA has the following optimization criteria:

$$L_{LDA} = \max_{\boldsymbol{W}} \frac{Tr\left(\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W}\right)}{Tr\left(\boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}\right)} \tag{13}$$

Equation (13) finds the projection transformation matrix $\boldsymbol{W}$ by the Lagrange multiplier method and then obtains the corresponding low-dimensional expression of $\boldsymbol{H}$ via $\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{H}$.

2.5. Locality Preserving Projections. As a linear transformation of the LE algorithm, the LPP algorithm solves the difficulty that the LE algorithm has in obtaining low-dimensional projection mapping on new test data [51] and is easily embedded by nonlinearity, thus finding a high-dimensional nonlinear manifold structure. LPP achieves dimensionality reduction by maintaining the neighbourhood structure of the data samples. LPP is obtained by linear transformation

$\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{H}$ on the basis of the LE algorithm. The LPP model can be expressed as follows:

$$\begin{aligned} L_{LPP} &= \min_{W} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2 \\ &= Tr\left(\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^T\right) \\ &= Tr\left(\left(\sum_{i=1}^{N} D_{ii}\boldsymbol{y}_i\boldsymbol{y}_i^T - \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}\boldsymbol{y}_i\boldsymbol{y}_j^T\right)\right) \\ &= Tr\left(\boldsymbol{Y}\left(\boldsymbol{D} - \boldsymbol{A}\right)\boldsymbol{Y}^T\right) \\ &= Tr\left(\boldsymbol{YLY}^T\right) = Tr\left(\boldsymbol{W}^T\boldsymbol{HLH}^T\boldsymbol{W}\right)_{LPP} \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{HDH}^T\boldsymbol{W} = \boldsymbol{I} \end{aligned} \tag{14}$$

In formula (14), $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d] \in R^{D\times d}$ is the projection transformation matrix. $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ represents the Laplacian matrix, where $\boldsymbol{D}$ is the diagonalization matrix, and $D_{ii} = \sum_{j=1}^{N} A_{ij}$. $\boldsymbol{A} \in R^{N\times N}$ is the sparse affinity matrix; if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are not near neighbours, then $A_{ij} = 0$. If $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are near neighbours, then $A_{ij} = \exp(-\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2/2\sigma^2)$. By learning a projection $\boldsymbol{W}$, the objective function minimizes the distance between those data points with neighbourhood relation in the raw data space.

3. Globality-Locality Preserving Maximum Variance ELM

3.1. Motivation of Globality-Locality Preserving Maximum Variance ELM. The local geometry of the sample can be used as side information for improving the performance of learning models. Assuming data samples $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are drawn from the same marginal distribution P_h, if two points $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are close to each other, then the conditional probabilities $P(\boldsymbol{y} \mid \boldsymbol{h}_1)$ and $P(\boldsymbol{y} \mid \boldsymbol{h}_2)$ should be similar as well. Based on local geometry of the sample, many locality preserving methods were proposed [56]. Zhao et al. proposed a new and effective semisupervised dimensionality reduction method, called Learning from Local and Global Information (LLGDI) [56], to utilize the underlying discriminative information. Literature [37] solves the problem that traditional subspace learning methods are the sensitivity to the outliers. They proposed a series of methods based on the L2,1-norm for dimensionality reduction. Literature [38] studies the problem that ridge regression based methods are sensitive to the variations of data and can learn only limited number of projections for feature extraction and recognition. They propose a new method called robust discriminant regression (RDR) for feature extraction. In literature [39], LLE and ONPP are combined to form the framework of sparse subspace learning. The framework is not only suitable for sparse linear subspace learning but also suitable for sparse nonlinear subspace learning. Essentially, our method can be viewed as one type of manifold learning, which is aimed at preserving the local geometry structure during feature learning or classification.

3.2. Manifold Regularization Framework. Manifold regularization framework can be obtained based on the LE algorithm [51]. However, because the LE algorithm has difficulty obtaining the low-dimensional projection mapping problem on the new test data [42], the LPP algorithm solves the above problems of the LE algorithm. Inspired by literature [40], based on the LPP algorithm, this paper proposes a manifold regularization framework. At the same time, considering that the LPP algorithm cannot maintain the global geometry of the data samples and the discriminant information contained in the data, this paper introduces the basic principles of the LDA algorithm into the manifold regularization framework. Compared with the literature [49, 50], the advantages of the algorithm proposed in this paper are as follows: (1) not only is the local manifold structure considered but also the global manifold structure and the global discriminant information of the data samples are considered; (2) taking into account the singularity of the manifold regularization framework, the maximum marginal criterion (MMC) [57] is used to solve the above problem. The LDA algorithm will make the similar samples closer but heterogeneous samples far away after the projection transformation. The LPP algorithm has advantage of maintaining the neighborhood structure of the sample after projection transformation. Therefore, the combined Section 2.4 and Section 2.5 manifold regularization framework loss function is shown in (14):

$$\begin{aligned} J &= \min_{W} Tr\left(\boldsymbol{W}^T\boldsymbol{HLH}^T\boldsymbol{W}\right)_{\text{LPP}} \\ &\quad - Tr\left(\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W} - \boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}\right)_{LDA} \\ &= Tr\left(\boldsymbol{W}^T\boldsymbol{HLH}^T\boldsymbol{W} - \boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W} + \boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}\right) \\ &= Tr\left(\boldsymbol{W}^T\left(\boldsymbol{HLH}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{W}\right) \end{aligned} \tag{15}$$

where $\boldsymbol{S}_W$ is an intraclass discrete matrix and $\boldsymbol{S}_B$ is an interclass discrete matrix as described in Section 2.4. $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d] \in R^{D\times d}$ is the projection transformation matrix and $\boldsymbol{I}$ is the unit matrix. $\boldsymbol{L}$ represents the Laplacian matrix and $\boldsymbol{D}$ is a diagonalization matrix as described in Section 2.5.

3.3. GLELM. The existing ELM algorithm cannot make good use of the intrinsic manifold structure information of the data, which can create the problem of insufficient learning. To overcome this problem, we propose a globality-locality preserving maximum extreme learning machine (GLELM) based on manifold learning. The optimization problem formulation of the GLELM is given by using the manifold regularization framework.

Based on manifold learning [9], assuming data samples $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are drawn from the same marginal distribution

P_h, if two points $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are close to each other, then the conditional probabilities $P(\boldsymbol{y} \mid \boldsymbol{h}_1)$ and $P(\boldsymbol{y} \mid \boldsymbol{h}_2)$ should be similar as well. The above assumption is widely known as the smoothness assumption in machine learning. In this subsection, we introduce the manifold regularization framework into the ELM model. In the ELM algorithm, $\boldsymbol{Y} = \boldsymbol{\beta}^T\boldsymbol{H}$; therefore, the GELM algorithm model can be written as follows:

$$\begin{aligned}
\min_{W} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\sum_{i=1}^{N}\|\boldsymbol{\varepsilon}_i\|^2 \\
& + \frac{1}{2}C_2\left(Tr\left(\boldsymbol{\beta}^T\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T\boldsymbol{\beta}\right)_{\text{LPP}} - Tr\left(\boldsymbol{\beta}^T\boldsymbol{S}_B\boldsymbol{\beta}\right.\right. \\
& \left.\left. - \boldsymbol{\beta}^T\boldsymbol{S}_W\boldsymbol{\beta}\right)_{LDA}\right) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\sum_{i=1}^{N}\|\boldsymbol{\varepsilon}_i\|^2 \\
& + C_2 Tr\left(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T\right. \\
& \left. - \boldsymbol{\beta}^T\boldsymbol{S}_B\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{S}_W\boldsymbol{\beta}\right) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\sum_{i=1}^{N}\|\boldsymbol{\varepsilon}_i\|^2 \\
& + C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T\right.\right. \\
& \left.\left. - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right) \\
s.t. \quad & \boldsymbol{\beta}^{\mathrm{T}}h\left(\boldsymbol{x}_i\right) = \boldsymbol{t}_i - \boldsymbol{\varepsilon}_i \\
& \qquad\qquad i = 1, 2, \cdots, N
\end{aligned} \tag{16}$$

where $Tr(\boldsymbol{\beta}^T\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T\boldsymbol{\beta})_{\text{LPP}} - Tr(\boldsymbol{\beta}^T\boldsymbol{S}_B\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{S}_W\boldsymbol{\beta})_{LDA}$ is the manifold regularization term, as described in Section 3.2. $\|\boldsymbol{\beta}\|^2$ is the l_2-norm regularization term; $\sum_{i=1}^{N}\|\boldsymbol{\varepsilon}_i\|^2$ is the training error term. C_1 is a penalty constant on the training errors, and C_2 is a penalty constant on the manifold regularization term. $h(\boldsymbol{x}_i) = (g(\boldsymbol{a}_1\cdot\boldsymbol{x}_i + b_1), g(\boldsymbol{a}_2\cdot\boldsymbol{x}_i + b_2), \ldots, g(\boldsymbol{a}_L\cdot\boldsymbol{x}_i + b_L))^T$ is the output vector of the hidden layer with respect to $\boldsymbol{x}_i$, as described in Section 2.3.

We rewrite (16) to the following form:

$$\begin{aligned}
J_{GLELM} = & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C_1\left\|\boldsymbol{\beta}^T\boldsymbol{H} - \boldsymbol{T}\right\|^2 \\
& + \frac{1}{2}C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right)
\end{aligned} \tag{17}$$

By substituting (17) in J_{GLELM} and solving for $\partial J_{GLELM}/\partial\boldsymbol{\beta} = 0$,

$$\begin{aligned}
\frac{\partial J_{GLELM}}{\partial\boldsymbol{\beta}} &= \frac{\partial\left((1/2)\|\boldsymbol{\beta}\|^2 + (1/2)C_1\left\|\boldsymbol{\beta}^T\boldsymbol{H} - \boldsymbol{T}\right\|^2 + (1/2)C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right)\right)}{\partial\boldsymbol{\beta}} \\
&= \frac{\partial\left((1/2)Tr\left(\boldsymbol{\beta}^T\boldsymbol{\beta}\right) + (1/2)C_1 Tr\left[\left(\boldsymbol{\beta}^T\boldsymbol{H} - \boldsymbol{T}\right)^T\left(\boldsymbol{\beta}^T\boldsymbol{H} - \boldsymbol{T}\right)\right] + (1/2)C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right)\right)}{\partial\boldsymbol{\beta}} \\
&= \frac{\partial\left((1/2)Tr\left(\boldsymbol{\beta}^T\boldsymbol{\beta}\right) + (1/2)C_1 Tr\left(\boldsymbol{H}^T\boldsymbol{\beta}\boldsymbol{\beta}^T\boldsymbol{H} - 2\boldsymbol{H}^T\boldsymbol{\beta}\boldsymbol{T}\right) + (1/2)C_2 Tr\left(\boldsymbol{\beta}^T\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}\right)\right)}{\partial\boldsymbol{\beta}} \\
&= \boldsymbol{\beta} + C_1\left(\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{\beta} - \boldsymbol{H}\boldsymbol{T}^T\right) + C_2\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta}
\end{aligned} \tag{18}$$

Order $\partial J_{GLELM}/\partial\boldsymbol{\beta} = 0$; according to formula (18), we can obtain the following formula:

$$\begin{aligned}
& \boldsymbol{\beta} + C_1\left(\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{\beta} - \boldsymbol{H}\boldsymbol{T}^T\right) + C_2\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\boldsymbol{\beta} \\
& \quad = 0
\end{aligned} \tag{19}$$

The output weight matrix $\boldsymbol{\beta}$ is obtained by solving formula (19) as follows:

$$\boldsymbol{\beta} = \left[\frac{\boldsymbol{I}}{C_1} + \boldsymbol{H}\boldsymbol{H}^T + \frac{C_2}{C_1}\left(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W\right)\right]^{-1}\boldsymbol{H}\boldsymbol{T}^T \tag{20}$$

Based on the above derivation, the specific steps of the GELLM algorithm are as shown in Algorithm 1.

4. Experiments

In this section, to verify the validity of the algorithm GLELM proposed in this paper, we use the image dataset and the UCI dataset [58] to perform experiments. A detailed description of the UCI dataset and image dataset is given in Table 1. In all our experiments, we compare the results of the GLELM experiments with the experimental results of ELM, MCVELM[44], RDELM[47], and GELM [49]. The specific comparison results are given in Figures 2 and 3 and Tables 2 and 3. For all ELMs, we choose the Sigmoid function as the activation function, and the number of hidden layer nodes is selected from $L = \{500, 1000\}$. For different ELM algorithms, on the training dataset, we use the threefold cross-validation and grid search methods to find the optimal parameters. For ELM, the MCVELM algorithm penalty parameter range is $C \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$. For RDELM, GELM and GLELM contain penalty parameters and regularization parameters, respectively, and the values are $C_1 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$ and $C_2 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$. In Figures 2 and 3, the abscissa "TrainNum" indicates the number of training samples for each category. In the experiment, we first randomly select

Input: Initialize the training sample set $K = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in R^d, \boldsymbol{t}_i \in R^m, i = 1, 2, \ldots, N\}$, activation function $g(x)$, the number of hidden layer nodes is L, the regularization parameters are C_1 and C_2;
Output: Output weight matrix $\boldsymbol{\beta}$;
Step 1: Randomly specify the network input weight $\boldsymbol{a}_j$ and offset value b_j, $j = 1, 2, \cdots, L$;
Step 2: Calculate the hidden layer node output matrix by the activation function $\boldsymbol{H}$;
Step 3: Calculate the manifold regularization framework according to formula (15) J;
Step 4: Calculate the output weight matrix from equation (20) $\boldsymbol{\beta}$.

Algorithm 1: GLELM algorithm.

Table 1: Dataset descriptions Dim is the number of feature dimensions, Samples is the number of data points, and Classes is the number of classes.

Datasets	Dim	Samples	Classes
ORL	1024	400	10
Yale	1024	165	15
Yale B	1024	2414	38
MNIST	784	70000	10
COIL20	1024	1440	20
USPS	256	9298	10
Iris	4	150	3
Wine	13	178	3
Segment	19	2310	7
Glass	9	214	6
Banknote	4	1372	2
Pima	8	768	2

Table 2: Average (sum average recognition rate of different training samples per subject) recognition rate of different ELM algorithms on image datasets (%).

Dataset	ELM Average	MCVELM Average	GELM Average	RDELM Average	GLELM Average
ORL	92.29	92.23	93.37	91.95	*93.87*
Yale	67.98	67.63	71.08	67.07	*75.05*
Yale B	95.74	96.04	97.05	97.17	*98.15*
MNIST	70.00	68.34	78.36	76.90	*83.49*
COIL20	90.74	93.35	94.33	94.26	*95.23*
USPS	87.08	89.37	91.10	91.18	*92.84*
Average	83.97	84.49	87.55	86.42	*89.77*

Table 3: Average (sum average recognition rate of different training samples per subject) recognition rate of different ELM algorithms on the UCI datasets (%).

Dataset	ELM Average	MCVELM Average	GELM Average	RDELM Average	GLELM Average
Iris	97.93	81.94	97.43	97.63	*98.37*
Wine	96.97	91.82	95.44	96.31	*97.69*
Segment	87.75	90.39	92.54	90.36	*94.26*
Glass	87.13	89.16	88.28	86.82	*91.93*
Banknote	99.74	99.55	99.68	99.65	*99.96*
Pima	59.77	60.60	62.46	60.82	*66.93*
Average	88.21	85.57	89.31	88.56	*91.52*

(a) ORL (b) Yale

(c) Yale B (d) COIL20

(e) MNIST (f) USPS

Figure 1: Different image datasets: (a) ORL, (b) Yale, (c) Yale B, (d) COIL20, (e) MNIST, and (f) USPS.

a part of the data from each type of data sample as the experimental dataset for the experiment. All experiments are conducted using a MATLAB 2015b computer with an Intel(R) Core (TM) 3.40 GHZ CPU and 8 GB RAM.

4.1. Datasets. To verify the classification performance of different algorithms, we use different types of datasets for the experiments, namely, handwritten digital image data, face image data, and the UCI datasets.

4.1.1. UCI Benchmark. We used six UCI machine learning datasets for the experiments. Table 1 presents details of the six UCI datasets we used in our experiments.

4.1.2. Image Datasets. We use a widely used image dataset as experimental data. The properties of the different image datasets are described below.

The Yale dataset [59] contains 165 images of 15 people, each containing 11 images. The images show the status of each person in different situations, such as happiness, sadness, and normality. The image size is 32×32.

The Yale B dataset [60] contains 2414 images of 38 people, each of which contains 55 images. The images show the state of each person in different situations, such as happiness, sadness, and normality. The image size is 32×32.

The ORL dataset [61] contains 400 images of 40 people. Each person contains 10 images. We selected images of different expressions under different lighting conditions. The image size is 32×32.

The COIL20 [62] dataset contains 1440 images of 20 types, each containing 72 images, and the image size is 32×32.

The MNIST dataset (http://yann.lecun.com/exdb/mnist/) is a handwritten digital image dataset containing 70,000 images with an image size of 28×28.

The USPS dataset [63] is a handwritten digital image containing 9298 images with an image size of 16×16.

Some samples of different image datasets are given in Figure 1. Table 1 gives specific details of the different image datasets.

4.2. Image Datasets Experiments. In this subsection, we show the experiments with different ELM algorithms on the image datasets. Figure 2 and Table 2 show the recognition rate curves and average recognition rates of different algorithms. In Figure 2, "TrainNum" denotes different training samples per subject. Figure 2 shows that the recognition rate curve of the GLELM algorithm in the six image datasets is better than the ELM, MCVELM, GELM, and RDELM algorithms. This is because the GLELM algorithm takes into account not only the local manifold structure information of the data samples but also the global geometry of the data samples and the discriminant information contained therein. The basic principle of the LDA and LPP algorithms is used to define a manifold regularization framework. At the same time, the manifold regularization framework is introduced into the ELM model to optimize the projection direction of the classification. As seen in Table 2, the recognition rate of the GELM algorithm on the ORL, Yale, Yale B, MNIST, COIL20, and USPS datasets is better than the ELM algorithm. This is because the GELM algorithm takes into account the local manifold information and discriminant information of the data samples and introduces the above information into the ELM model to optimize the output weights and enhance the classification performance. The experimental results of the MCVELM algorithm on the ORL and Yale dataset are very close to those of the ELM algorithm. The recognition rates on the Yale B, COIL20, and USPS datasets are better than those of the ELM algorithm. This is because the MCVELM algorithm takes into account the discriminant information of the data samples, thereby introducing the intraclass divergence matrix into the ELM model. The recognition rate of the RDELM algorithm on the Yale B, MNIST, COIL20, and USPS datasets is better than the ELM algorithm. This is because the RDELM algorithm takes into account the intraclass information and interclass discrimination information of the data samples. The intraclass divergence matrix and the interclass divergence matrix are introduced into the ELM model.

4.3. UCI Datasets Experiments. To further verify the effectiveness of the proposed GLELM algorithm, we conduct experiments on the UCI dataset by GLELM, ELM, MCVELM, and GELM. Figure 2 and Table 2 show the recognition rate curves and average recognition rates of different ELM algorithms. In Figure 3, "TrainNum" denotes different training samples per subject. Figure 3 shows that the recognition rate curve of the GLELM algorithm in the six UCI datasets is better than the ELM, MCVELM, GELM, and RDELM algorithms. The recognition rate of the MCVELM

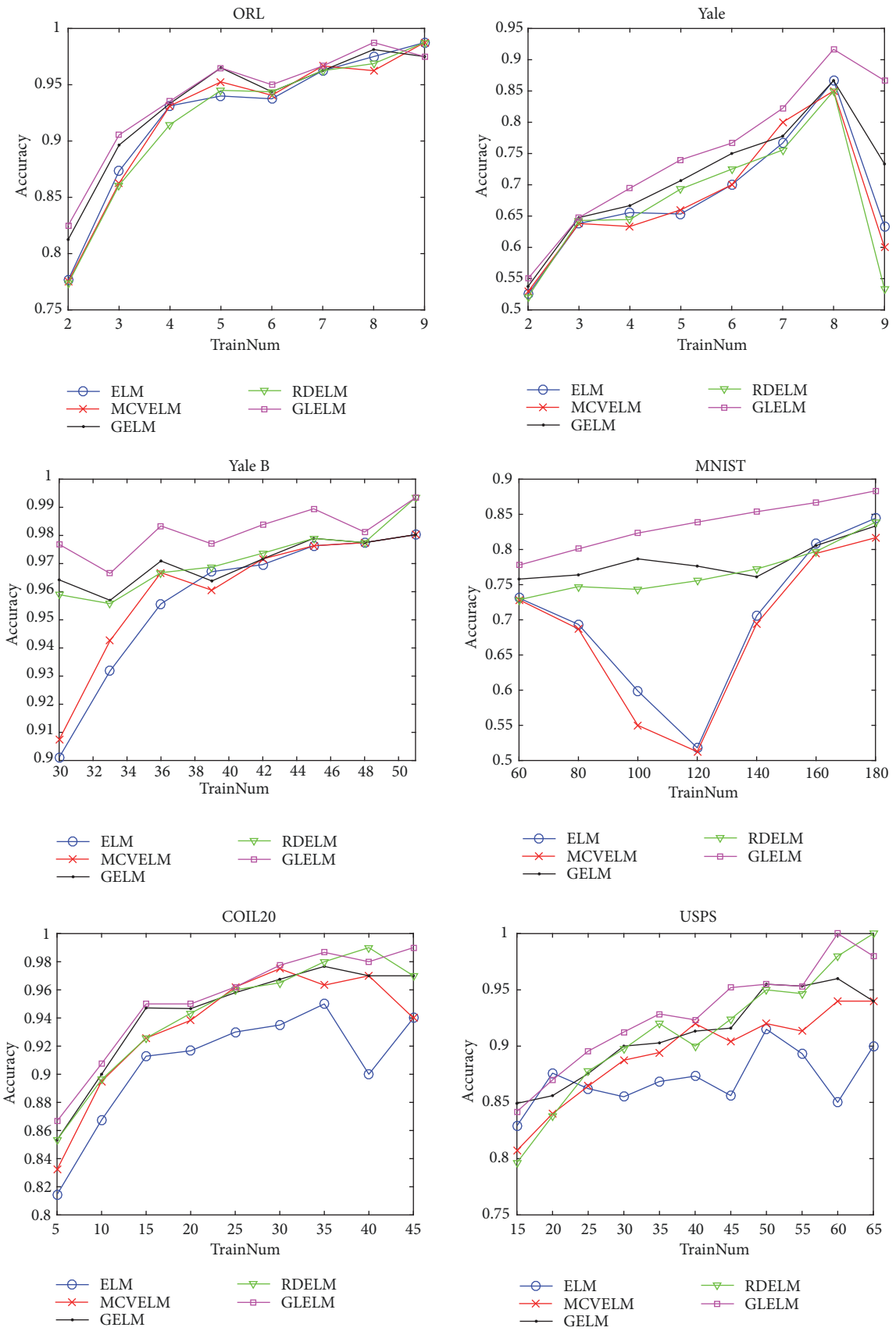

Figure 2: Recognition rate curves of different ELM algorithms on image datasets ORL, Yale, Yale B, COIL20, MNIST, and USPS.

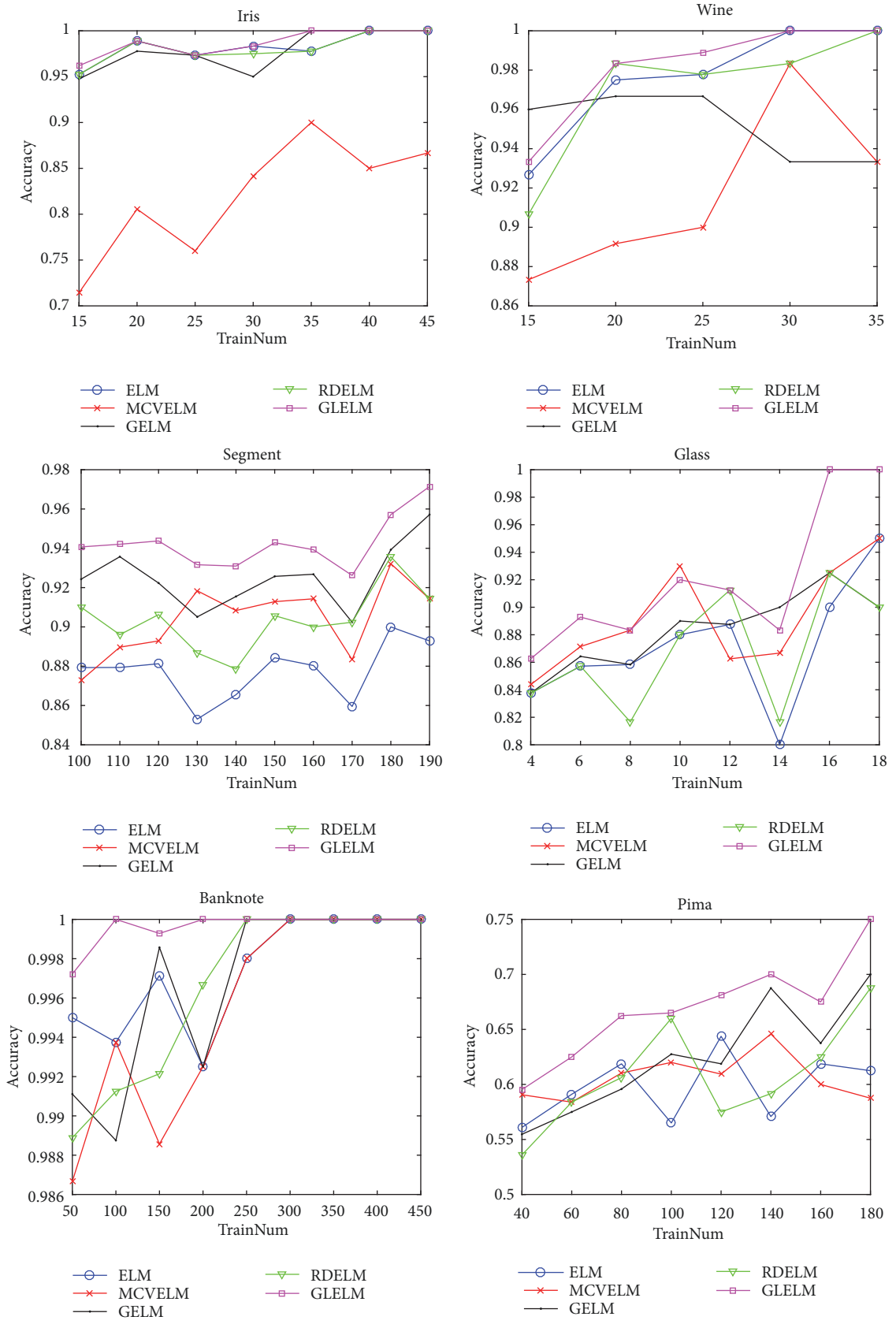

Figure 3: Identification rate curves of different ELM algorithms on UCI datasets Iris, Wine, Segment, Glass, Banknote, and Pima.

Table 4: Average training time for different ELM algorithms on UCI datasets and images (training time)(s).

Dataset	ELM Average	MCVELM Average	GELM Average	RDELM Average	GLELM Average
ORL	0.3291	2.7930	1.9805	2.7500	2.8828
Yale	1.1904	1.8145	1.3643	1.7773	1.9258
Yale B	1.7256	3.4258	26.6641	3.5576	4.4004
MNIST	1.4855	1.8996	13.4877	2.0480	2.5569
USPS	1.2862	1.4858	3.0412	1.7109	1.8942
COIL20	1.2890	1.9280	4.3394	2.0825	2.4965
Iris	0.1752	0.1842	0.2321	0.2478	0.3281
Wine	0.2281	0.2547	0.2766	0.2844	0.3547
Segment	0.2813	0.3867	7.5508	0.4672	0.7695
Glass	0.1797	0.1953	0.1709	0.2295	0.3203
Banknote	0.2648	0.2700	1.8924	0.3385	0.3941
Pima	0.2031	0.2285	0.4961	0.2520	0.3496

and GELM algorithms on the Segment, Glass, and Pima datasets is better than that of the ELM algorithm. MCVELM, GELM, and ELM are very close to the Banknote dataset. Based on Figures 2 and 3, we can see that the difference in the recognition rate curve of different ELM algorithms on the image dataset is relatively small. However, the recognition rate curves of different ELM algorithms on the UCI dataset fluctuate greatly, which may have a certain relationship with the dimensionality of the data. The dimensionality of the image data is generally small, while the UCI dataset dimension is generally large.

The experimental results of different ELM algorithms on the image dataset and the UCI dataset show that the proposed GLELM algorithm enhances the classification performance and generalization ability of the ELM algorithm by introducing a manifold regularization framework in the ELM model.

4.4. Parameter Analysis for GLELM. The two parameters included in the ELM algorithm are the number of hidden layer nodes L and the penalty parameter C. The GLELM algorithm contains three parameters, namely, the number of hidden layer nodes L, the penalty parameters C_1, and regularization parameters C_2. Based on the literature [7], the performance of ELM is not very sensitive to the number of hidden nodes. When performing experiments on the UCI dataset, we set the hidden layer node $L = 500$ and set the hidden layer node to $L = 1000$ on the image dataset. We conduct experiments to investigate the effect of these parameters on the final recognition accuracy penalty parameters C_1 and regularization parameters C_2; the values are $C_1 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$ and $C_2 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$. We use the image datasets to perform experiments to observe the effect of different parameter values on the final recognition rate. In the experiments, for the ORL and Yale datasets, we select the number of data samples in each subclass as 5 for the training set and the remainder as the test set. For the Yale B and MNIST datasets, the number of data samples is selected as the training set in each subclass, and the remaining data are used as the test set. For the COIL20 and USPS datasets, the number of data samples is selected as the training set in each subclass, and the remaining data are used as the test set. On the image dataset, the effect of different values of penalty parameter C_1 and regularization parameter C_2 on the GLELM algorithm is shown in Figure 4.

4.5. Computing Complexity Analysis. In this subsection, we analyse the computational complexity of different algorithms, for the ELM algorithm $\boldsymbol{\beta} = (\boldsymbol{I}/C + \boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{T}^T$, $\boldsymbol{H}\boldsymbol{H}^T$ is a matrix of $L \times L$ and L is the number of hidden layer nodes. In most cases, the number of hidden layer nodes L is much smaller than the number of training samples N (i.e., $L \ll N$). Thus, the computational cost decreases dramatically compared to LS-SVM and PSVM, which needs to compute the inverse of an $N \times N$ matrix [24]. Similarly, our proposed GLELM, RDELM, GELM, and MCVELM have similar complexity as conventional ELM. RDELM output weights can be written as $\boldsymbol{\beta} = (\boldsymbol{I}/C_2 + \boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{S}/C_1)^{-1}\boldsymbol{H}\boldsymbol{T}^T$; $\boldsymbol{S}$ is the difference between the intraclass divergence matrix and the interclass divergence matrix. The output weight of GELM $\boldsymbol{\beta} = (C_2\boldsymbol{I} + \boldsymbol{H}\boldsymbol{H}^T + C_1\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{T}^T$, the output weight of GLELM $\boldsymbol{\beta} = [\boldsymbol{I}/C_1 + \boldsymbol{H}\boldsymbol{H}^T + C_2/C_1(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T - \boldsymbol{S}_B + \boldsymbol{S}_W)]^{-1}\boldsymbol{H}\boldsymbol{T}^T$, the output weight of MCVELM $\boldsymbol{\beta} = [\boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{S}/C]^{-1}\boldsymbol{H}\boldsymbol{T}^T$ $\boldsymbol{S}$ are the intraclass divergence matrix. All of the algorithms need to calculate the $L \times L$ inverse matrix of HH^T. Therefore, the computational complexity of the above different ELM algorithms is $O(L^3)$. Figure 5 shows the training time for different ELM algorithms on the image dataset and the UCI dataset.

Figure 5 shows the training time curves of different ELM algorithms on image datasets and UCI datasets. Table 4 shows the average training time for different ELM algorithms. As shown in Table 4, the average training time of the four algorithms MCVELM, GELM, RDELM, and GLELM on the image dataset and the UCI dataset is higher than the average training time of the ELM algorithm. This is because the four algorithms MCVELM, GELM, RDELM, and GLELM all introduce regular terms based on the ELM algorithm. From Table 4, we observe that the average training time of the GLELM algorithm on the ORL, Yale, Iris, Wine and

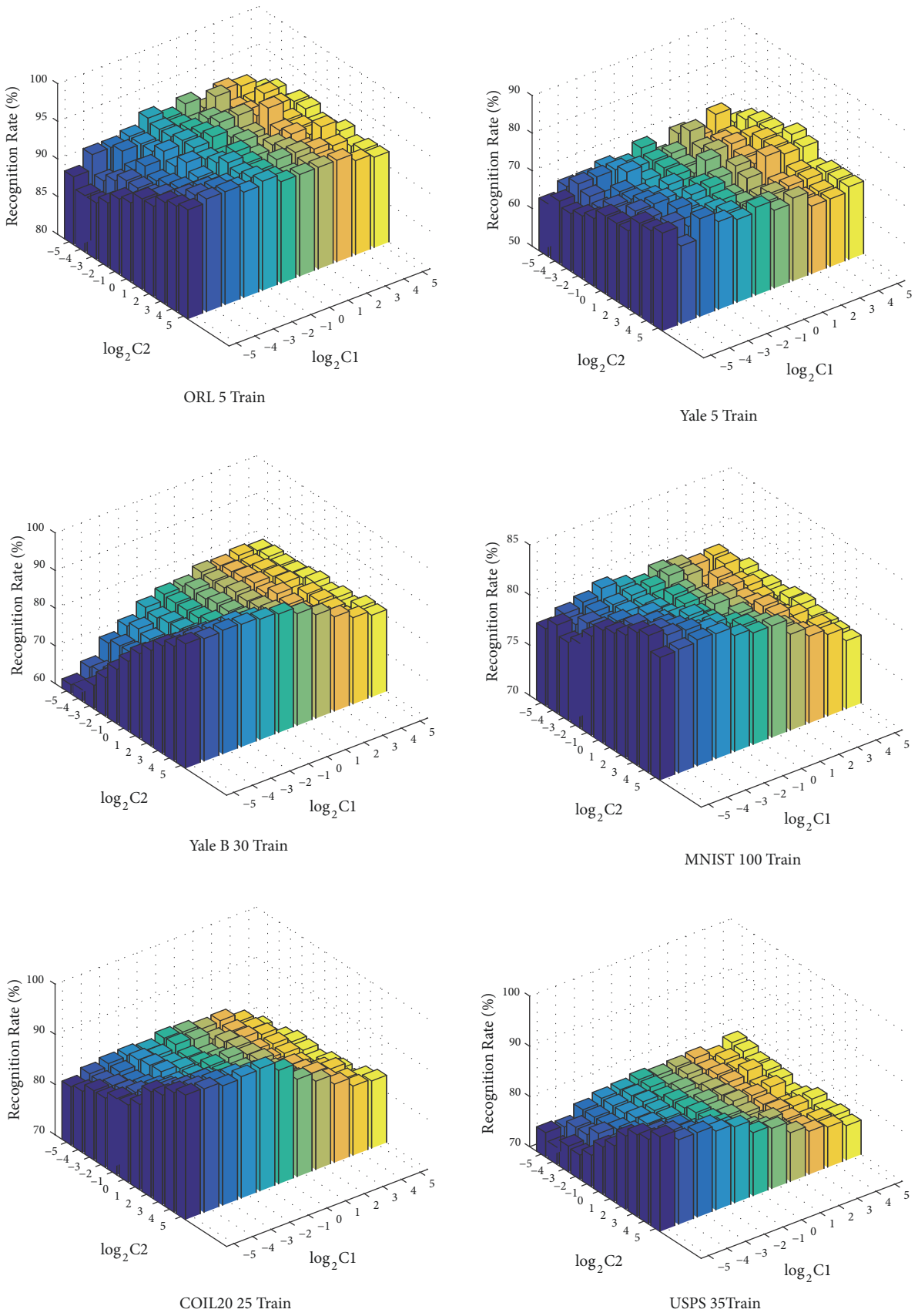

FIGURE 4: Penalty parameters C_1 and regularization parameters C_2 on the image dataset. The effect of different values on the GELLM algorithm, ORL, Yale, Yale B, COIL20, MNIST, and USPS.

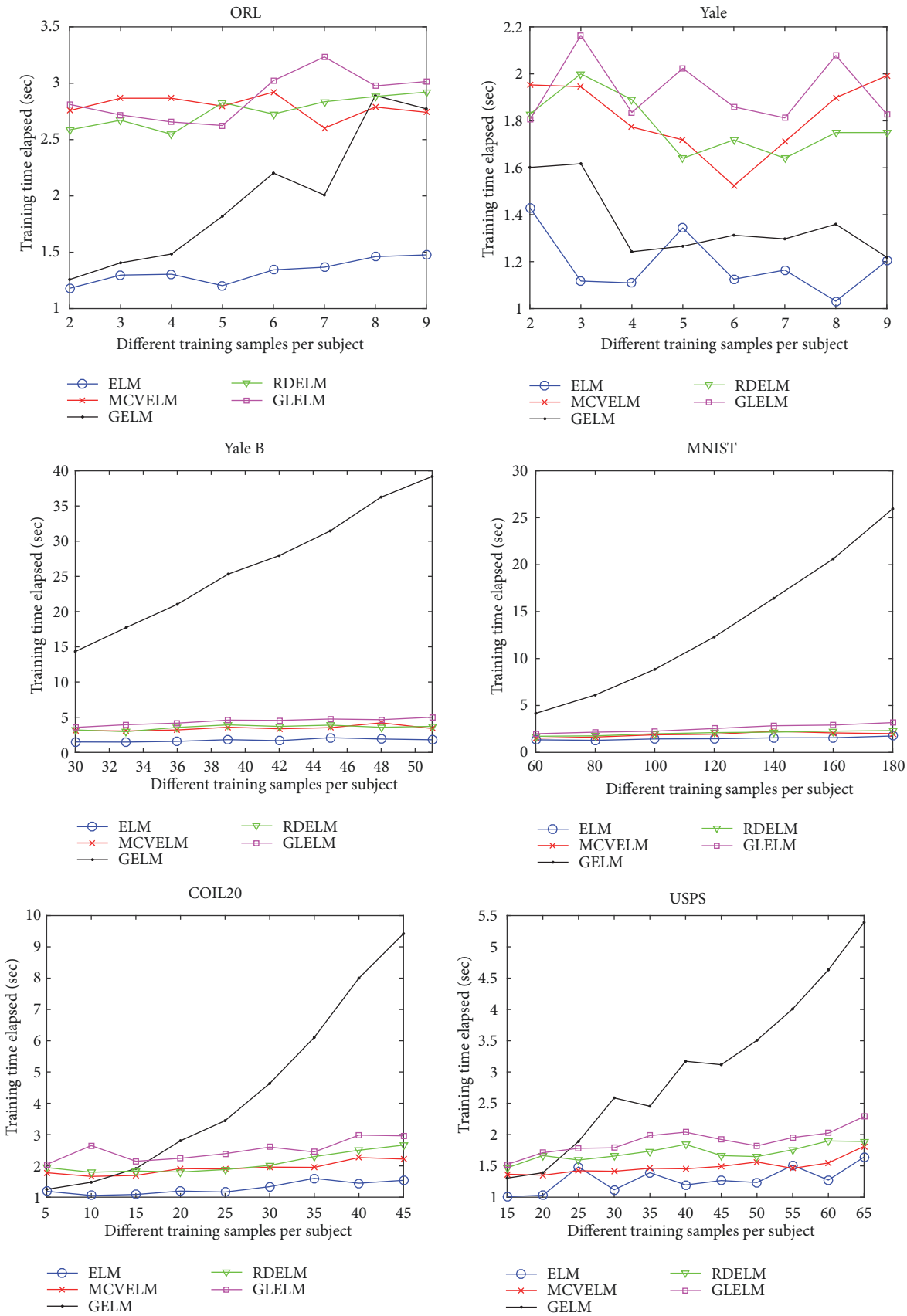

Figure 5: Continued.

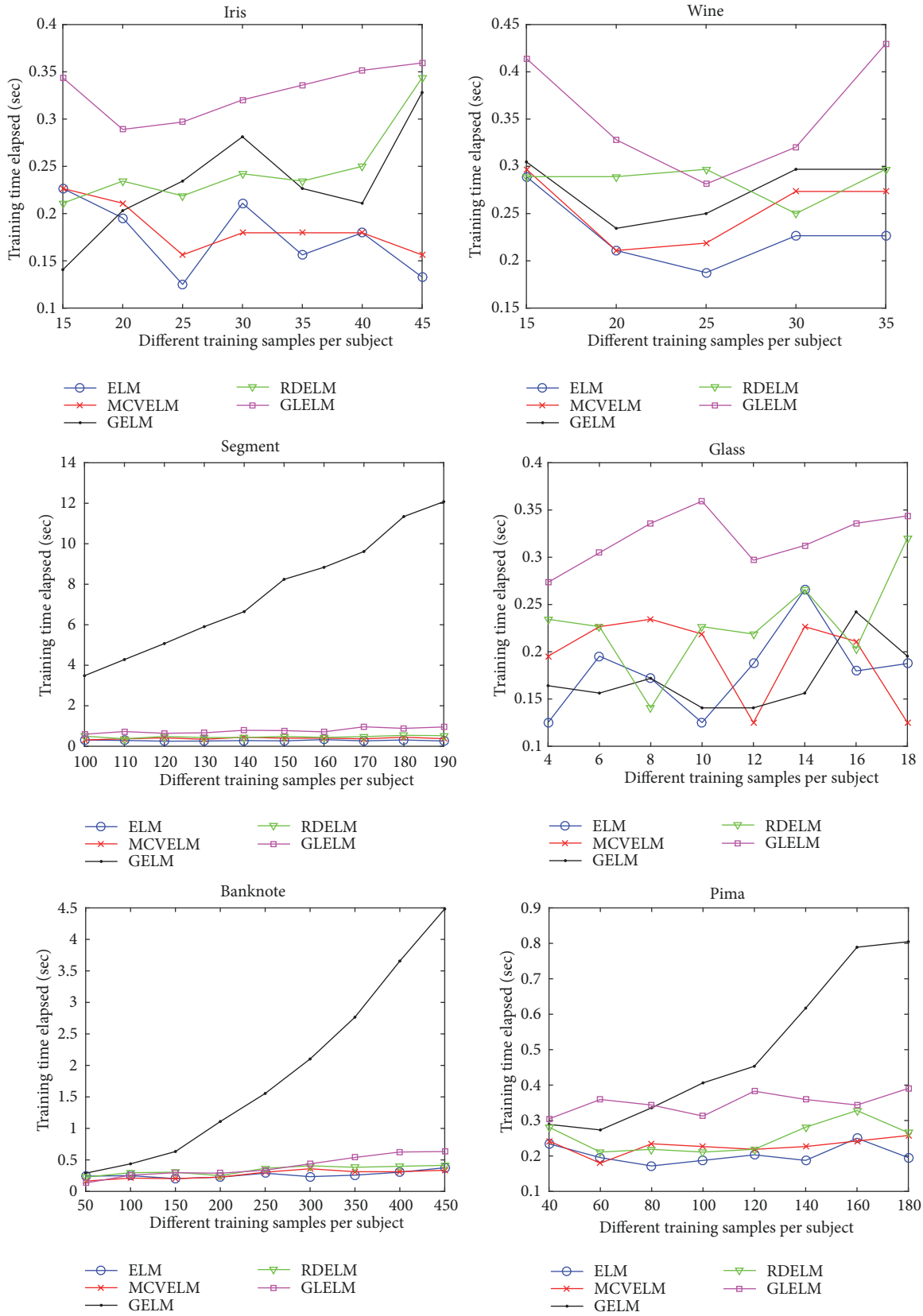

Figure 5: Different ELM algorithm training time curves on UCI datasets and image datasets Iris, Wine, Segment, Glass, Banknote, Pima, ORL, Yale, Yale B, COIL20, MNIST, and USPS.

Table 5: Classification accuracy (mean± std%) of different methods with different numbers of training samples on the AR database.

Algorithms	AR			
	11 Train	14 Train	17 Train	20 Train
LCVELM	94.65±0.58	96.23±0.53	97.43±0.46	98.82±0.45
GEELM	96.57±0.56	97.28±0.77	98.52±0.44	98.87±0.63
GLELM	*97.13±0.58*	*98.16±0.43*	*98.81±0.26*	*99.02±0.47*

Table 6: Classification accuracy (mean± std%) of different methods with different numbers of training samples on the Isolet1 database.

Algorithms	Isolet1			
	5 Train	8 Train	11 Train	14 Train
LCVELM	81,35±1.37	85.66±1.02	86.94±1.12	88.11±0.45
GEELM	81.98±1.52	86.56±1.00	88.48±1.12	90.07±0.63
GLELM	*83.17±1.05*	*87.77±0.68*	*89.68±0.83*	*91.23±0.47*

Glass datasets is higher than that of other algorithms. The average training time of GELM on eight datasets of Segment, Banknote, Pima, Yale B, COIL20, MNIST, and USPS is higher than that of other ELM algorithms.

Based on the analysis of the 12 datasets in Figure 5 and Table 4, we find that Segment, Banknote, Pima, Yale B, COIL20, MNIST, and USPS contain considerably more training samples than ORL, Yale, Iris, Wine, and Glass. The aforementioned phenomenon may be caused by time consumption of GELM, in which more adjacency graph matrix to be constructed as sample data increases. From this, we can infer that when the dataset contains a large number of training samples, the time efficiency of GLELM will be better than that of the GELM algorithm.

Combining Tables 2, 3, and 4, we analyse the performance of the proposed algorithm GLELM in terms of the classification recognition rate and time efficiency. Tables 2 and 3 show that GLELM has good classification ability and is superior to other ELM algorithms. As shown in Table 4, GLELM does not have a considerable advantage on time overhead, and it is not the least time-consuming algorithm on some datasets. However, based on the ELM classification speed and good classification performance, GLELM can be used as an effective classifier in pattern recognition.

5. Relationship between LCVEM and GEELM Models as well as GLELM

In this section, we analyze the proposed algorithms GLELM compared with LCVELM and GEELM in detail. Combining the discriminant information of the data sample and the adjacency graph structure, the LCVELM algorithm constructs Laplacian feature matrix to obtain the local geometric structure of the data sample and strengthen the generalization ability of the ELM algorithm. We design the graph embedding framework based on the LCVELM algorithm in LCVELM algorithm, which can be composed of algorithms such as LLE algorithm, LE algorithm, and LDA algorithm. The GEELM algorithm constructs the graph embedding framework based on the LCVELM algorithm. The graph embedding framework can be composed of algorithms such as LLE algorithm, LE algorithm, and LDA algorithm. These steps have enabled GEELM to further improve the classification capabilities of LCVELM. The GGLEL algorithm takes both the local and global geometry and the discriminant information of the data into consideration, which differ from the LCVELM algorithm and the GEELM algorithm essentially only using the local geometry of the data samples. In addition, different manifold regularization structures make GLELM, GEELM, and GLELM have distinguishing classification performance. In order to verify the classification performance of the three algorithms, we conduct experiments on AR face image datasets and speech data. For all ELMs, we choose the Sigmoid function as the activation function, and the number of hidden layer nodes is selected from $L = 1000$. For different ELM algorithms, on the training dataset, we use the threefold cross-validation and grid search methods to find the optimal parameters. For LCVELM, GEELM and GLELM contain penalty parameters and regularization parameters, respectively, and the values are $C_1 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$ and $C_2 \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$. For the AR dataset, we randomly select $l_{AR} = \{11, 14, 17, 20\}$ images per subject for training and the rest for testing. Similarly, for the remaining datasets, we set $l_{Isolet1} = \{5, 8, 11, 14\}$. All experiments were randomly selected from the training set and the test set to run 10 times, and then the average of the 10 runs was calculated as the final recognition result. The recognition results of different ELM algorithms on the image dataset are shown in Tables 5 and 6.

(i) The AR face database [64]: dataset contains 4,000 images of 126 people (70 men and 56 women). We use a subset that contains 2600 grey images of 100 human subjects and have selected images of different expressions under different lighting conditions. In our experiments, the images were cropped and scaled to an image size of 50×40.

(ii) Isolet1 (http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html): It contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers each and are referred to as isolet1 through isolet5. We selected Isolet1 for the experiment.

Tables 5 and 6 show the comparison result of the recognition rates of LCVELM, GEELM, and GELLM on the AR face image dataset and the Isolet1 voice dataset. From Tables 5 and 6, we can see that the recognition rate of GELEM algorithm is better than LCVELM and GEELM algorithm. Through the experimental results, we can conclude that the local geometric information and global geometry of the data sample can effectively enhance the recognition effect of the ELM algorithm and is better than the LCVELM and GEELM algorithm.

6. Conclusions

Currently ELM faces the research hot point that ELM algorithm cannot fully use the local geometric structure and the global geometric structure information of data samples. Although researchers have made efforts to solve this problem by proposing related algorithms as MCVELM, GELM, and GEELM, these algorithms give a way in different sides; the more effective research is expected. In order to describe the geometry of the data sample we introduce manifold learning into the ELM algorithm. In manifold learning, LPP can depict local geometrical structure well. LDA can acquire the global geometric structure of data samples and the discriminant information as well. We adopt the basic principles of the LDA and LPP algorithms, define a manifold regular framework, introduce the manifold regular framework into the ELM model, and propose a globality-locality preserving extreme learning machine algorithm (GLELM). Compared with ELM, GLELM can acquire manifold structure information of samples and is of the stronger ability to recognize. We validate the GLELM algorithm using image datasets and UCI datasets, and the experimental results verify the effectiveness of GLELM. In the future, We will introduce local discriminant information of data into the ELM algorithm, to explore the effect on the performance of recognition.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by a grant from the Natural Science Foundation of China (nos. 61632011, 61572102, 61702080, and 61602079) and the Fundamental Research Funds for the Central Universities (nos. DUT18ZD102 and DUT17RC(3)016).

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[2] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 6, pp. 989–993, 1994.

[3] J. Branke, "Evolutionary algorithms for neural network design and training," in *Proceedings of the First Nordic Workshop on Genetic Algorithms and Its Applications*, 1995.

[4] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.

[5] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[6] G. Huang, L. Chen, and C. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 4, pp. 879–892, 2006.

[7] R. Zhang, Y. Lan, G.-B. Huang, and Z.-B. Xu, "Universal approximation of extreme learning machine with adaptive growth of hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 365–371, 2012.

[8] S. Barro and J. Ribeiro, "Direct kernel perceptron (DKP): Ultra-fast kernel ELM-based classification with noniterative closed-form weight calculation," *Neural Networks*, vol. 50, pp. 60–71, 2014.

[9] G. Huang and S. J. Song, "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*, vol. 8, Springer, 6 edition, 1986.

[11] X. Liu and L. Xu, "The universal consistency of extreme learning machine," *Neurocomputing*, vol. 311, pp. 176–182, 2018.

[12] Z. Ren and L. Yang, "Correntropy-based robust extreme learning machine for classification," *Neurocomputing*, vol. 313, pp. 74–84, 2018.

[13] S. Li, S. Song, and Y. Wan, "Laplacian twin extreme learning machine for semi-supervised classification," *Neurocomputing*, vol. 321, pp. 17–27, 2018.

[14] B. S. Raghuwanshi and S. Shukla, "Class-specific kernelized extreme learning machine for binary class imbalance learning," *Applied Soft Computing*, vol. 73, pp. 1026–1038, 2018.

[15] Y. Zhang, J. Wu, C. Zhou, and Z. Cai, "Instance cloned extreme learning machine," *Pattern Recognition*, vol. 68, pp. 52–65, 2017.

[16] X. Liu, S. B. Lin, J. Fang, and Z. B. Xu, "Is extreme learning machine feasible? a theoretical assessment (part II)," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 7–20, 2015.

[17] X. Liu, C. Gao, and P. Li, "A comparative analysis of support vector machines and extreme learning machines," *Neural Networks*, vol. 33, pp. 58–66, 2012.

[18] S. B. Lin, X. Liu, J. Fang, and Z. B. Xu, "Is extreme learning machine feasible? a theoretical assessment (part I)," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 21–34, 2015.

[19] X. Z. Wang, Q. Y. Shao, Q. Miao, and J. H. Zhai, "Architecture selection for networks trained with extreme learning machine using localized generalization error model," *Neurocomputing*, vol. 102, pp. 3–9, 2013.

[20] L.-C. Shi and B.-L. Lu, "EEG-based vigilance estimation using extreme learning machines," *Neurocomputing*, vol. 102, pp. 135–143, 2013.

[21] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

[22] E. H. Zheng, C. Zhang, X. Y Liu, H. J. Lu, and J. Sun, "Cost-Sensitive Extreme Learning Machine," in *Proceedings of the International Conference on Advanced Data Mining and Applications*, pp. 478–488, 2013.

[23] R. Annalisa, F. N. Francisco, and C. Sante, "Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1898–1909, 2014.

[24] L. Zhang and D. Zhang, "Evolutionary cost-sensitive extreme learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 3045–3060, 2017.

[25] K. Li, X. F. Kong, Z. Lu, and W. Y. Liu, "Boosting weighted ELM for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, 2014.

[26] Y. C. Zhou and J. T. Peng, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2351–2360, 2015.

[27] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, "E^2LMs: ensemble extreme learning machines for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1060–1069, 2014.

[28] L. F. Javier and Q. B. Pablo, "Efficient ELM-based techniques for the classification of hyperspectral remote sensing images on commodity GPUs," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2884–2893, 2015.

[29] X.-R. Zhou and C.-S. Wang, "Cholesky factorization based online regularized and kernelized extreme learning machines with forgetting mechanism," *Neurocomputing*, vol. 174, pp. 1147–1155, 2016.

[30] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, "PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis," *Neural Processing Letters*, vol. 37, no. 3, pp. 377–392, 2013.

[31] Q. Wang, W. G. Wang, R. Nian, and B. He, "Manifold learning in local tangent space via extreme learning machine," *Neurocomputing*, vol. 174, pp. 18–30, 2016.

[32] P. B. Zhang and Z. X. Yang, "A robust AdaBoost.RT based ensemble extreme learning machine," *Mathematical Problems in Engineering*, vol. 2015, Article ID 260970, 12 pages, 2015.

[33] N. Liu and H. Wang, "Ensemble based extreme learning machine," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 754–757, 2010.

[34] G. Deepak and L. Joonwhoan, "Extreme learning machine ensemble using bagging for facial expression recognition," *Journal of Information Processing Systems*, vol. 10, no. 3, pp. 443–458, 2014.

[35] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Mining on manifolds: metric learning without labels," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7642–7651, Salt Lake City, UT, June 2018.

[36] S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto, "Zero shot learning via multi-Scale manifold regularization," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 5292–5299, USA, July 2017.

[37] Z. Lai, Y. Xu, J. Yang, L. Shen, and D. Zhang, "Rotational invariant dimensionality reduction algorithms," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3733–3746, 2017.

[38] Z. H. Lai, D. M. Mo, W. K. Wong, and Y. Xu, "Robust discriminant regression for feature extraction," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2472–2484, 2018.

[39] Z. H. Lai, D. M. Mo, W. K. Wong, and Y. Xu, "Approximate orthogonal sparse embedding for dimensionality reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 723–735, 2016.

[40] J. Gui, W. Jia, L. Zhu, S.-L. Wang, and D.-S. Huang, "Locality preserving discriminant projections for face and palmprint recognition," *Neurocomputing*, vol. 73, no. 13, pp. 2696–2707, 2010.

[41] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[42] X. F. He and P. Niyogi, "Locality preserving projections," *Advances in Neural Information Processing Systems*, vol. 16, pp. 153–160, 2004.

[43] X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi, and H. J. Zhang, "Face recognition using laplacian faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[44] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1968–1979, 2013.

[45] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum Variance Extreme Learning Machine for human action recognition," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2014*, pp. 5427–5431, Italy, May 2014.

[46] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.

[47] S. Liu, L. Feng, Y. Liu, J. Wu, M. X. Sun, and W. Wang, "Robust discriminative extreme learning machine for relevance feedback in image retrieval," *Multidimensional Systems and Signal Processing*, vol. 28, no. 3, pp. 1071–1089, 2017.

[48] A. Iosifidis, A. Tefas, and P. Ioannis, "Exploiting local class information in extreme learning machine," in *Proceedings of the 6th International Conference on Neural Computation Theory and Applications, NCTA 2014, Part of the 6th International Joint Conference on Computational Intelligence, IJCCI 2014*, pp. 49–55, Italy, October 2014.

[49] Y. Peng and B. L. Lu, "Discriminative graph regularized extreme learning machine and its application to face recognition," *Neurocomputing*, pp. 340–353, 2015.

[50] Y. Peng and B.-L. Lu, "Discriminative manifold extreme learning machine and applications to image and EEG signal classification," *Neurocomputing*, vol. 174, pp. 265–277, 2014.

[51] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[52] H. X. Wang, S. B. Chen, Z. L. Hu, and W. M. Zheng, "Locality-preserved maximum information projection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 4, pp. 571–585, 2008.

[53] S. Huang, A. Elgammal, L. Huangfu, D. Yang, and X. Zhang, "Globality-locality preserving projections for biometric data dimensionality reduction," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2014*, pp. 15–20, USA, June 2014.

[54] S. Huang, A. Elgammal, J. Lu, and D. Yang, "Cross-speed gait recognition using speed-invariant gait templates and globality-locality preserving projections," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2071–2083, 2015.

[55] C. L. Ma and Y. B. Yuan, "A novel support vector machine with globality-locality preserving," *The Scientific World Journal*, vol. 2014, Article ID 872697, 6 pages, 2014.

[56] M. Zhao, T. W. S. Chow, Z. Wu, Z. Zhang, and B. Li, "Learning from normalized local and global discriminative information for semi-supervised regression and dimensionality reduction," *Information Sciences*, vol. 324, pp. 286–309, 2015.

[57] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 1, pp. 157–165, 2006.

[58] K. Bache and M. Lichman, *UCI Machine Learning Repository*, School of Information and Computer Sciences, University California, Irvine, CA, USA, 2013, http://archive.ics.uci.edu/ml.

[59] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'07*, USA, June 2007.

[60] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, Rio de Janeiro, Brazil, October 2007.

[61] D. Cai, X. F. He, J. W. Han, and H. J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.

[62] D. Cai, X. F. He, J. W. Han, and T. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

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

[64] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.