Neurocomputing 74 (2011) 2541–2551

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Face recognition based on extreme learning machine

Weiwei Zong, Guang-Bin Huang *

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Available online 26 May 2011

Keywords:
Face recognition
Discriminative locality alignment
Support vector machine
Extreme learning machine
One-against-all
One-against-one

ABSTRACT

Extreme learning machine (ELM) is an efficient learning algorithm for generalized single hidden layer feedforward networks (SLFNs), which performs well in both regression and classification applications. It has recently been shown that from the optimization point of view ELM and support vector machine (SVM) are equivalent but ELM has less stringent optimization constraints. Due to the mild optimization constraints ELM can be easy of implementation and usually obtains better generalization performance. In this paper we study the performance of the one-against-all (OAA) and one-against-one (OAO) ELM for classification in multi-label face recognition applications. The performance is verified through four benchmarking face image data sets.

1. Introduction

As one of the most important biometric methods, face recognition has been attracting much attention from researchers [1,2]. In general, there are two main steps in a face recognition system. As shown in Fig. 1, the first step is to define an effective representation of the face images, which includes sufficient information of the face for future classification. The second step is to classify a new face image with the chosen representation. The approaches used to represent a face image can be mainly divided into two categories [3,4]: holistic-based and feature-based. The holistic-based methods (such as Eigenfaces [5], Fisherfaces [6]) take the whole face region as the raw input into a recognition system. The feature-based methods extract the local features of the eyes, nose, and mouth first, and then collect the locations and local statistics of those features for classification. In this paper, active appearance model (AAM) [7] will be used as the feature-based learning method. AAM iteratively matches a statistical model on new images by making use of the correlations of the model parameter and the corresponding residual errors. In this paper, AAM is used as a feature extraction tool to extract the local feature information, that is, the shape, texture and appearance of every image.

Since natural images usually contain significant statistical redundancies, dimensionality reduction is adopted in both holistic-based and feature-based methods [2], which transforms the image data from high-dimensional space to low-dimensional space in order to reduce redundancies and noises as well as to reveal the essential features of face images. Many dimensionality reduction algorithms have been proposed and studied in the literature [2,4,8]. Among them, principal components analysis (PCA) [2] and linear discriminant analysis (LDA) [2] have played important roles as the classic approaches for dimensionality reduction. In PCA, by decomposing the covariance matrix for input vectors, the inputs can be expressed as linear combinations of the orthogonal basis by solving an eigen problem. Although the data in the original space may be reconstructed well, PCA is not an optimal choice for classification problems [9]. By utilizing the class label information, the projection directions of LDA are determined to maximize the ratio of between-class deviation and within-class deviation. However, there exist some potential problems [10]: the nonlinear structure in the data is not revealed; and the matrix singularity problem may happen. Discriminative locality alignment (DLA) [10] is proposed to overcome those problems. In DLA, a patch for each sample is built. This patch includes the sample itself and the neighbor samples from the same class as well as from different classes. First, the discriminative information of each patch is preserved by an objective function. Then, the processed local patches are integrated to compute the final solution.

Nearest neighbor (NN) classifier and its variants [2] are widely used in different algorithms such as PCA-based Eigenfaces [5] and LDA-based Fisherfaces [6]. An NN classifier simply calculates the distance between the testing image and every training image, and fixes the identity whose images are closest to the testing image. Supervised classifiers like support vector machine (SVM) [11] and neural network [1] are also proposed in face recognition system. The SVM is originally proposed as a binary classifier. For multi-class applications, researchers have developed one-against-all SVM (SVM-OAA) [4] and one-against-one SVM (SVM-OAO) [12]. For an *M*-label classification application, SVM-OAA is composed of *M* SVMs with each SVM distinguishing only one class of samples

* Corresponding author.
E-mail address: egbhuang@ntu.edu.sg (G.-B. Huang).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.12.041

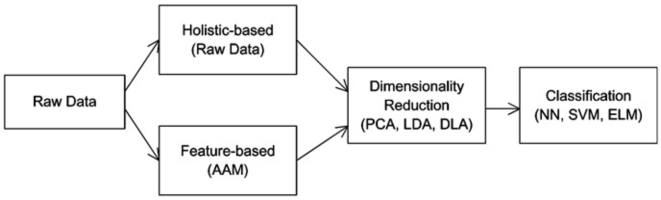

Fig. 1. The main procedures in a face recognition system.

from the rest classes of samples; SVM-OAO consists of $M(M-1)/2$ SVMs with each SVM distinguishing two classes of samples. Although there are a few face recognition applications using the combination of SVM and subspace learning methods [13–15], to the best of our knowledge there is no study on the combination of SVM and DLA till now.

Huang et al. [16,17] proposed a new learning algorithm called extreme learning machine (ELM) for single hidden layer feedforward networks (SLFNs) which can be used in regression and classification applications [18–21]. ELM runs fast and is easy to implement. ELM randomly generates the hidden node parameters, and then analytically determines the output weights of SLFNs instead of iterative tuning. ELM has been successfully applied to palmprint recognition [22]. Recently, Huang et al. [23] applied the optimization method in finding the solution to ELM in the classification applications. It was found that the optimization method-based ELM for classification and support vector network are actually equivalent but ELM has milder optimization constraints, and thus ELM for classification can be easy of implementation. ELM for classification usually achieves better generalization performance without intensive human intervene.

In this paper we propose the one-against-all ELM (ELM-OAA) and one-against-one ELM (ELM-OAO) as classifiers in the application of face recognition. The performance of ELM-OAO/ELM-OAA is compared to SVM-OAA/SVM-OAO with both holistic learning and feature-based learning on four benchmarking face data sets.

2. Brief of dimensionality reduction methods

This section provides a brief review on the dimensionality reduction methods to be used with our proposed classifiers ELM-OAA and ELM-OAO. In our experiments, to demonstrate the proposed classifiers, we use both holistic learning and feature-based learning methods to represent the facial images, followed by different dimensionality reduction methods including the classic PCA [2,5] and LDA [2,24], and DLA recently proposed by Zhang et al. [10].

2.1. Principal component analysis

PCA [2,5] has been widely used as a classic dimensionality reduction method. Given an input vector $\mathbf{x}$, the following eigen problem can be computed:

$$\mathbf{C}\boldsymbol{\Phi} = \lambda\boldsymbol{\Phi} \tag{1}$$

where $\mathbf{C}$ is the covariance matrix for $\mathbf{x}$. $\boldsymbol{\Phi}$ and λ represent eigen vector and eigen value of this equation, respectively. $\mathbf{x}$ can be viewed as a linear combinations of $\boldsymbol{\Phi}$ obtained above:

$$\mathbf{x} = \sum_{i=1}^{n} a_i \boldsymbol{\Phi}_i \tag{2}$$

where n is the number of eigen vectors obtained from (1), and $\mathbf{x}$ can be represented by the coefficients a_i, $i=1,\ldots,n$. Since the rank of covariance matrix in (1) is bounded by the number of training samples (N), the dimensionality of samples (referred to as P-dim) after PCA projection will not exceed N.

2.2. Linear discriminant analysis

During LDA training [2,24], different from PCA, the label information (ω_i, $i=1,\ldots,M$) of the input $\mathbf{x}$ with M classes is utilized by computing the within-class scatter matrix $\mathbf{S}_w$ and between-class scatter matrix $\mathbf{S}_b$:

$$\mathbf{S}_w = \sum_{i=1}^{M} Pr(\omega_i)\mathbf{C}_i$$

$$\mathbf{S}_b = \sum_{i=1}^{M} Pr(\omega_i)(\mathbf{m}_i - \mathbf{m}_0)(\mathbf{m}_i - \mathbf{m}_0)^T \tag{3}$$

where $\mathbf{C}_i$ is the covariance matrix. $Pr(\omega_i)$ is the prior class probability for input vectors belonging to class ω_i which is usually replaced by $1/M$ [2]. $\mathbf{m}_i$ and $\mathbf{m}_0$ represent the mean for input vectors from class ω_i and the global mean of all input vectors from all classes, respectively.

To maximize the ratio of the determinants of $\mathbf{S}_b$ and $\mathbf{S}_w$, we compute the projection matrix $\mathbf{W}$ corresponding to the eigen value λ_W of the generalized eigen problem:

$$\mathbf{S}_b\mathbf{W} = \lambda_W \mathbf{S}_w \mathbf{W} \tag{4}$$

Hence, after projection, the samples from different classes are farther apart and the samples from the same class become closer. The dimensionality of samples after LDA projection is not larger than (M–1) where M is the number of classes since the rank of S_b is bounded by (M–1) [25].

2.3. Discriminative locality alignment

In LDA, the nonlinear structure in the data is not revealed and the matrix singularity problem may happen. As a linear dimensionality reduction method, DLA [10] is proposed to overcome the particular problems in LDA.

Suppose that we have a set of samples $X=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$, each sample $\boldsymbol{x}_i \in \mathbf{R}^n$ labeled with one of the M labels. For a given sample $\boldsymbol{x}_i$, with reference to the label information, k_1 neighbor samples from the same class are randomly selected: $\boldsymbol{x}_{i^1},\ldots,\boldsymbol{x}_{i^{k_1}}$. Another k_2 nearest neighbors for $\boldsymbol{x}_i$ are randomly selected from samples in different classes: $\boldsymbol{x}_{i_1},\ldots,\boldsymbol{x}_{i_{k_2}}$. Then the local patch $\mathbf{X}_i$ for sample $\boldsymbol{x}_i$ is formed as follows:

$$\mathbf{X}_i = [\boldsymbol{x}_i, \boldsymbol{x}_{i^1}, \ldots, \boldsymbol{x}_{i^{k_1}}, \boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{k_2}}]$$

For each patch, after transformation, the local patch becomes

$$\mathbf{Y}_i = [\boldsymbol{y}_i, \boldsymbol{y}_{i^1}, \ldots, \boldsymbol{y}_{i^{k_1}}, \boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_{k_2}}]$$

The goal is to make sure that after transformation, the given sample is close to samples from the same class and far away from the samples in different classes in terms of Euclidean distance. The mathematical form is:

$$\arg\min_{\boldsymbol{y}_i} \left(\sum_{j=1}^{k_1} \|\boldsymbol{y}_i - \boldsymbol{y}_{i^j}\|^2 - C_{DLA} \sum_{p=1}^{k_2} \|\boldsymbol{y}_i - \boldsymbol{y}_{i_p}\|^2 \right) \tag{5}$$

where $C_{DLA} \in [0,1]$ is defined by the user to justify the proportion of within-class distance and between-class distance. If we denote

$$\boldsymbol{w}_i = \left[\overbrace{1,\ldots,1}^{k_1}, \overbrace{-C_{DLA},\ldots,-C_{DLA}}^{k_2} \right]^T$$

(5) then becomes:

$$\arg\min_{\mathbf{Y}_i} \operatorname{tr}(\mathbf{Y}_i\mathbf{L}_i\mathbf{Y}_i^T) \tag{6}$$

where

$$\mathbf{L}_i = \begin{bmatrix} \sum_{j=1}^{k_1+k_2}(\boldsymbol{w}_i)_j & -\boldsymbol{w}_i^T \\ -\boldsymbol{w}_i & \operatorname{diag}(\boldsymbol{w}_i) \end{bmatrix}$$

The details of the deduction can be found in [10]. It can be found that different from LDA, by focusing on the local patch of each sample so that not only the discriminative information but also the non-linearity of the sample distribution are preserved.

For each sample, the optimization for its local patch ($\mathbf{X}_i$, $i=1,\ldots,N$) is described as in (6). To unify the optimizations of all the samples as a whole, a new definition called *selection matrix* is introduced. For the ith patch $\mathbf{Y}_i = [\boldsymbol{y}_i, \boldsymbol{y}_{i^1}, \ldots \boldsymbol{y}_{i^{k_1}}, \boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_{k_2}}]$, $\mathbf{F}_i = \{i, i^1, \ldots, i^{k_1}, i_1, \ldots, i_{k_2}\}$ is the corresponding index set, and the selection matrix $\mathbf{S}_i$ is

$$(\mathbf{S}_i)_{pq} = \begin{cases} 1 & \text{if } p = \mathbf{F}_i\{q\} \\ 0 & \text{else} \end{cases}$$

where $\mathbf{S}_i \in \mathbf{R}^{N\times(k_1+k_2+1)}$, $p=1,\ldots,N$ and $q=1,\ldots,k_1+k_2+1$. Thus, $\mathbf{Y}_i$ can be expressed as $\mathbf{YS}_i$, where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ is the collection of the local patches for all the N samples. Substituting it into (6), we have:

$$\arg\min_{\mathbf{Y}} \sum_{i=1}^{N} \operatorname{tr}(\mathbf{YS}_i\mathbf{L}_i\mathbf{S}_i^T\mathbf{Y}^T) = \arg\min_{\mathbf{Y}} \operatorname{tr}(\mathbf{YLY}^T) \tag{7}$$

where $\mathbf{L} = \sum_{i=1}^{N} \mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T \in \mathbf{R}^{N\times N}$ is the alignment matrix.

By imposing $\mathbf{U}^T\mathbf{U} = \mathbf{I}_d$, where $\mathbf{U}$ is the transformation matrix and $\mathbf{I}_d$ is a $d\times d$ identity matrix, (7) can be rewritten as:

$$\arg\min_{\mathbf{U}} \operatorname{tr}(\mathbf{U}^T\mathbf{XLX}^T\mathbf{U}). \tag{8}$$

Finally the solution of (8) and the transformation matrix can be obtained by operating the standard eigen-decomposition of $\mathbf{XLX}^T\boldsymbol{u} = \lambda\boldsymbol{u}$, which is free of matrix singularity problem since we need not compute the matrix inverse as is the case for the eigen equation used in LDA.

3. Multi-label classifiers

3.1. Brief of support vector machine

Given a set of training data $(\mathbf{x}_i, t_i)$, $i=1,\ldots,N$, where $\mathbf{x}_i \in \mathbf{R}^d$ and $t_i \in \{-1,1\}$, SVM [26] aims to find a hyperplane to separate the data into two classes with maximum separating margin after mapping these data from the input space to the feature space by $\phi(\mathbf{x})$: $\mathbf{x}_i \to \phi(\mathbf{x}_i)$.

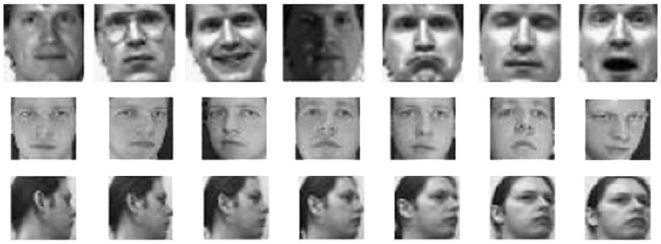

Fig. 2. Face samples: Row 1 for YALE, Row 2 for ORL, Row 3 for UMIST.

To maximize the separating margin distance as well as to minimize the training error we have the following optimization function:

$$\begin{aligned} &\text{Minimize}: \quad L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{Subject to}: \quad t_i(\mathbf{w}\cdot\phi(\mathbf{x}_i)+b) \geq 1-\xi_i, \quad i=1,\ldots,N, \\ &\qquad \xi_i \geq 0, \quad i=1,\ldots,N \end{aligned} \tag{9}$$

where C is a user specified parameter.

According to the Karush–Kuhn–Tucker theorem [27], the optimization problem is equivalent to

$$\begin{aligned} &\text{minimize}: \quad L_D = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t_it_j\alpha_i\alpha_j\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i \\ &\text{subject to}: \quad \sum_{i=1}^{N} t_i\alpha_i = 0 \\ &\quad 0 \leq \alpha_i \leq C, \quad i=1,\ldots,N \end{aligned} \tag{10}$$

where each Lagrange multiplier α_i corresponds to a training example $(\mathbf{x}_i, t_i)$.

With the introduction of the kernel function $K(\mathbf{u},\mathbf{v}) = \phi(\mathbf{u})\cdot\phi(\mathbf{v})$, the optimization function can be rewritten as:

$$\begin{aligned} &\text{minimize}: \quad L_D = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t_it_jK(\mathbf{x}_i,\mathbf{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \\ &\text{subject to}: \quad \sum_{i=1}^{N} t_i\alpha_i = 0 \\ &\quad 0 \leq \alpha_i \leq C, \quad i=1,\ldots,N, \end{aligned} \tag{11}$$

3.2. Brief of extreme learning machine

Extreme learning machine [16,17] is a learning method for generalized single hidden layer feedforward networks where the hidden nodes can be neuron alike [28] and non-neuron alike [29,30]. The output of an ELM is

$$f(\mathbf{x}) = \sum_{i=1}^{L}\beta_iG(\mathbf{a}_i,b_i,\mathbf{x}) = \boldsymbol{\beta}\cdot\mathbf{h}(\mathbf{x}), \tag{12}$$

where $\mathbf{h}(\mathbf{x}) = [G(\mathbf{a}_1,b_1,\mathbf{x}),\ldots,G(\mathbf{a}_L,b_L,\mathbf{x})]$ is the output vector for the hidden layer with respect to input $\mathbf{x}$. The parameters for hidden layer nodes are randomly assigned and the output weight β_i which connects the ith hidden node to the output nodes is then analytically determined.

Similar to SVM, we have:

$$\begin{aligned} &\text{Minimize}: \quad \sum_{i=1}^{N}\|\boldsymbol{\beta}\cdot\mathbf{h}(\mathbf{x}_i)-t_i\| \\ &\quad \text{and} \\ &\text{Minimize}: \quad \|\boldsymbol{\beta}\|^2. \end{aligned} \tag{13}$$

ELM is to minimize the training errors as well as the norm of output weights $\|\boldsymbol{\beta}\|^2$.

Fig. 3. Face samples from BioID database.

3.3. Brief of ELM for classification

As can be seen through the last two subsections, the similarity and relationship between ELM and SVM from optimization point of view have been pointed out by Huang et al. [23]. In [23], ELM can be reformulated as

$$\text{Minimize}: \quad L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{Subject to}: \quad t_i\boldsymbol{\beta}\cdot\mathbf{h}(\mathbf{x}_i) \geq 1-\xi_i, \quad i=1,\ldots,N$$
$$\xi_i \geq 0, \quad i=1,\ldots,N, \tag{14}$$

where $\mathbf{h}(\mathbf{x})$ is the ELM mapping for input $\mathbf{x}$. Hence, similar optimization function and constraints for ELM can be drawn according to (9). The difference between ELM and SVM lies in the bias b. Different from SVM, ELM does not have the bias b since the separating hyperplane in ELM feature space tends to pass the origin.

The ELM kernel is defined as [23]

$$K_{\text{ELM}}(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{h}(\mathbf{x}_i)\cdot\mathbf{h}(\mathbf{x}_j) = \sum_{s=1}^{L} G(\mathbf{a}_s,b_s,\mathbf{x}_i)G(\mathbf{a}_s,b_s,\mathbf{x}_j) \tag{15}$$

The corresponding dual optimization function can be obtained as

$$\text{minimize}: \quad L_D = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t_it_jK_{\text{ELM}}(\mathbf{x}_i,\mathbf{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i$$
$$\text{subject to}: \quad 0 \leq \alpha_i \leq C, \quad i=1,\ldots,N. \tag{16}$$

It can be seen that ELM for classification is essentially a combination of ELM kernel and standard optimization method which is used to find the solution. Comparing (11) and (16) it is found that ELM has milder optimization constraints. Compared with SVM, ELM does not have the optimization condition: $\sum_{i=1}^{N} t_i\alpha_i = 0$. Thus, the solution of SVM is suboptimal compared to ELM.

Table 1
The optimal parameter (P-dim, k_1, k_2, C_{DLA}) of DLA in different systems.

	NN	SVM-OAA	SVM-OAO	ELM-OAA	ELM-OAO
UMIST (7)	(33,4,5,0.1)	(25,5,4,0.1)	(21,5,1,0.1)	(26,5,2,0.1)	(26,5,2,0.1)
UMIST (5)	(10,3,4,0.1)	(24,3,3,0.1)	(31,2,1,0.1)	(28,3,3,0.1)	(28,3,3,0.1)
UMIST (3)	(18,2,1,0.1)	(21,2,1,0.1)	(18,2,1,0.1)	(18,2,1,0.1)	(18,2,1,0.1)
ORL (7)	(35,2,4,0.3)	(40,3,4,0.1)	(45,3,8,0.1)	(45,3,3,0.1)	(45,3,3,0.1)
ORL (5)	(30,3,2,0.8)	(40,4,4,0.1)	(40,3,8,0.1)	(39,4,2,0.1)	(39,4,2,0.5)
ORL (3)	(30,2,4,0.1)	(30,2,4,0.1)	(36,2,5,0.1)	(32,2,5,0.1)	(32,2,5,0.1)
YALE (7)	(15,3,5,0.1)	(14,3,4,0.1)	(15,4,2,0.3)	(15,3,4,0.1)	(15,3,4,0.1)
YALE (5)	(30,3,4,0.1)	(14,3,3,0.1)	(14,3,3,0.1)	(14,3,6,0.1)	(14,3,3,0.1)
YALE (3)	(18,2,1,0.1)	(14,2,1,0.3)	(14,2,1,0.3)	(14,2,1,0.5)	(14,2,1,0.1)
BioID (5)	(21,4,5,0.1)	(25,4,7,0.1)	(25,3,7,0.1)	(25,4,6,0.1)	(25,4,4,0.3)
BioID (4)	(21,3,2,0.1)	(25,3,3,0.1)	(25,3,3,0.1)	(30,3,3,0.1)	(25,3,3,0.1)
BioID (3)	(21,2,2,0.1)	(25,2,3,0.1)	(25,2,1,0.1)	(27,2,2,0.1)	(27,2,3,0.1)
BioID (2)	(21,1,1,0.1)	(25,1,3,0.1)	(25,1,3,0.1)	(27,1,3,0.1)	(25,1,3,0.1)

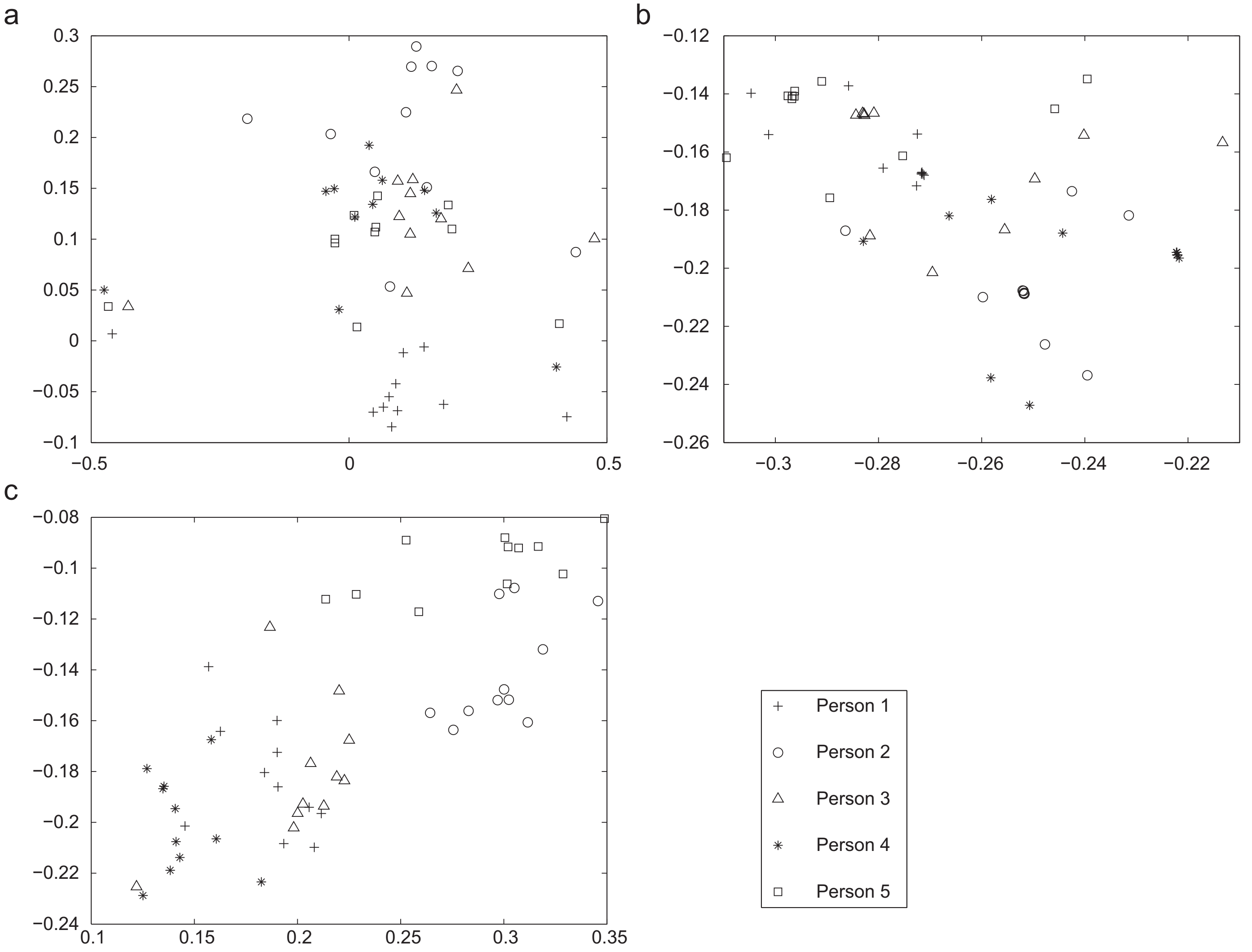

Fig. 4. Two-dimensional representation of the images of selected individuals from YALE. (a) PCA(5), (b) LDA(5) and (c) DLA(5).

3.4. One-against-all and one-against-one for SVM and ELM

Seen from (14), ELM for classification works as a binary classifier. In multi-label applications, researchers have proposed different kinds of solutions. Among those methods, one-against-all (OAA) [4] and one-against-one (OAO) [12] are popular and widely used.

Seen from Eq. (16) the decision function of ELM for classification is

$$f(\mathbf{x}) = \text{sign}\left(\sum_{s=1}^{N_s} \alpha_s t_s K_{\text{ELM}}(\mathbf{x},\mathbf{x}_s)\right) \tag{17}$$

Given the multi-label data from M different classes, OAA consists of M binary classifiers with each one trained to distinguish each class and the rest classes. For one testing data, the classifier with the largest output of decision function wins the competition and the corresponding class is assigned to the testing data [31]. To the best of our knowledge, OAA appears to be the most popular method for multi-class SVM classification nowadays [13–15].

In OAO, one binary classifier is used to distinguish one pair of classes, which results in $(M-1)*M/2$ binary classifiers in total. For each classifier, the sign of the output of decision function will indicate which class from that particular pair has been chosen. For

Table 2
Comparison between the NN, SVM-OAA, SVM-OAO, ELM-OAA and ELM-OAO on database YALE.

YALE	NN		OAA	SVM						ELM				
(P-dim)	Testing rate (%)	Testing dev (%)	or OAO	Testing rate (%)	Testing dev (%)	Training time (s)	C	γ	# SVs	Testing rate (%)	Testing dev (%)	Training time (s)	C	L
PCA (7)	60.50	5.41	OAA	81.83	4.98	0.75	10^4	10	418	82.08	4.25	2.21	50	1000
(105)			OAO	76.00	4.57	2.30	10^4	10	1163	76.50	3.97	2.15	2	1000
PCA (5)	56.33	3.77	OAA	76.61	4.64	0.67	10^4	20	335	76.50	4.13	1.41	1000	1000
(75)			OAO	69.89	3.96	2.05	10^4	10	872	69.50	4.01	1.63	20	1000
PCA (3)	48.46	3.99	OAA	66.83	4.29	0.50	10^3	5	251	66.79	4.56	0.48	5	1000
(45)			OAO	58.92	4.45	1.64	10^3	5	594	58.71	5.05	1.02	100	1000
LDA (7)	83.83	6.21	OAA	85.33	4.61	0.52	10^{-3}	0.1	326	83.75	4.95	1.24	10	1000
(14)			OAO	85.00	5.97	2.10	1	0.2	1045	84.67	5.50	2.36	2	1000
LDA (5)	78.89	3.69	OAA	80.89	3.67	0.48	10^{-3}	0.1	358	79.78	4.53	1.07	2	1000
(14)			OAO	79.61	3.33	1.72	1	0.1	653	79.33	3.94	2.04	2	1000
LDA (3)	64.33	4.77	OAA	68.00	4.30	0.50	1	0.2	239	66.04	4.47	0.41	2	1000
(14)			OAO	65.08	5.41	1.53	2	0.2	522	64.67	5.59	0.68	2	1000
DLA (7)	87.25	3.68	OAA	89.00	3.03	1.13	10^4	10	156	89.04	2.70	1.68	5	1000
			OAO	89.17	2.73	3.25	10^4	0.1	616	88.67	3.04	2.02	2	1000
DLA (5)	82.00	3.94	OAA	83.61	3.12	1.28	10^4	0.1	323	83.63	2.91	1.74	1	1000
			OAO	82.67	3.91	5.56	10^4	0.1	530	83.22	3.38	1.44	1	1000
DLA (3)	70.37	3.79	OAA	72.54	3.17	1.04	10^4	0.1	409	72.50	4.12	0.78	2	1000
			OAO	71.21	3.86	4.60	10^4	0.1	573	70.71	4.16	0.85	20	1000

Table 3
Comparison between the NN, SVM-OAA, SVM-OAO, ELM-OAA and ELM-OAO on database UMIST.

UMIST	NN		OAA	SVM						ELM				
(P-dim)	Testing rate (%)	Testing dev (%)	or OAO	Testing rate (%)	Testing dev (%)	Training time (s)	C	γ	# SVs	Testing rate (%)	Testing dev (%)	Training time (s)	C	L
PCA (7)	88.91	2.54	OAA	93.15	2.18	1.49	50	0.8	588	92.55	2.09	2.09	10	1000
(140)			OAO	93.32	2.53	7.12	50	2	1980	93.49	2.25	3.19	100	1000
PCA (5)	80.84	3.51	OAA	87.66	3.05	0.97	50	0.8	463	87.33	2.90	1.34	50	1000
(100)			OAO	86.87	3.05	7.23	10^4	10	1570	87.27	2.41	2.44	100	1000
PCA (3)	67.53	3.36	OAA	75.26	3.90	0.38	50	1	343	75.21	3.81	0.67	10^3	1000
(60)			OAO	73.49	4.54	5.44	10^4	10	1088	73.70	3.45	1.52	50	1000
LDA (7)	92.05	2.30	OAA	92.01	2.32	1.24	0.5	0.1	451	91.28	2.41	1.29	5	1000
(19)			OAO	92.13	2.30	3.27	5	0.1	637	92.35	2.82	1.15	50	1000
LDA (5)	86.67	2.51	OAA	87.35	2.54	0.70	0.2	0.1	493	85.92	2.86	1.14	1	1000
(19)			OAO	86.87	2.48	4.73	1	0.2	1534	86.76	1.99	1.06	100	1000
LDA (3)	75.37	4.60	OAA	75.91	4.58	0.62	0.5	0.1	752	74.49	4.62	0.53	10	1000
(19)			OAO	75.52	4.43	5.22	50	0.4	742	75.74	3.87	1.03	10^3	1000
DLA(7)	96.09	1.50	OAA	96.22	1.35	3.18	10^{-3}	0.1	551	94.66	1.84	4.45	2	1000
			OAO	96.01	1.43	6.18	10^4	1	414	96.17	1.65	1.23	10	1000
DLA (5)	91.76	2.22	OAA	92.24	2.39	1.56	10^{-3}	0.1	605	89.96	2.48	1.12	1	1000
			OAO	91.93	2.33	6.18	10^4	10	453	91.88	2.22	1.08	10	1000
DLA (3)	80.85	3.88	OAA	81.52	3.78	0.51	10^{-3}	0.1	518	78.55	3.96	2.02	2	1000
			OAO	80.85	3.90	6.02	10^3	0.1	948	80.77	3.93	0.93	10	1000

testing data, the simplest way to determine the class is by selecting the one chosen by maximum numbers of classifiers, which we use in this paper. Advanced methods including decision graphs [32] have been proposed in the literature.

4. Performance evaluation

In this section we study the performance of ELM-OAA and ELM-OAO in the combination with holistic learning method as

Table 4
Comparison between the NN, SVM-OAA, SVM-OAO, ELM-OAA and ELM-OAO on database ORL.

ORL	NN		OAA	SVM						ELM				
(*P*-dim)	Testing rate (%)	Testing dev (%)	or OAO	Testing rate (%)	Testing dev (%)	Training time (s)	*C*	γ	# SVs	Testing rate (%)	Testing dev (%)	Training time (s)	*C*	*L*
PCA (7)	91.67	2.37	OAA	97.46	1.39	23.2	10	0.4	1484	97.11	1.63	6.38	10^3	1000
(280)			OAO	96.67	1.79	65.8	10	0.4	8509	96.28	2.42	20.1	5	1000
PCA (5)	86.25	1.57	OAA	94.65	1.86	3.14	50	1	1158	94.97	1.47	3.84	20	1000
(200)			OAO	92.90	1.82	52.5	50	0.8	6459	92.40	1.78	14.7	5	1000
PCA (3)	77.29	3.31	OAA	88.09	2.49	2.72	10	0.4	1084	88.29	2.35	2.04	5	1000
(120)			OAO	83.59	3.26	28.8	50	0.8	4396	83.57	3.09	10.4	10	1000
LDA (7)	95.50	1.63	OAA	97.04	1.22	153	10	10^4	11200	96.46	1.38	7.68	5	1000
(39)			OAO	95.87	1.82	21.3	50	0.8	8362	95.62	1.60	8.05	5	1000
LDA (5)	92.77	1.97	OAA	94.57	1.96	24.7	10	10^4	8000	93.93	2.23	9.28	1	1000
(39)			OAO	93.15	2.13	18.8	10	0.2	3364	92.75	1.78	5.98	10	1000
LDA (3)	84.59	3.74	OAA	88.09	2.95	1.20	10	0.4	967	87.50	3.12	2.74	20	1000
(39)			OAO	84.09	3.40	20.2	50	0.8	3914	84.02	3.80	4.21	50	1000
DLA(7)	97.62	1.49	OAA	98.58	1.33	0.55	10^4	0.1	3652	98.44	1.17	4.54	5	1000
			OAO	98.50	1.29	86.6	10^4	10	3637	98.54	1.57	5.91	50	1000
DLA (5)	96.20	1.75	OAA	97.20	1.49	1.18	10^4	0.1	2552	96.73	1.21	1.52	10	1000
			OAO	96.80	1.62	12.5	10^4	0.1	4055	96.65	1.71	26.4	10	1000
DLA (3)	90.14	2.58	OAA	91.38	2.93	2.01	10^4	0.1	3057	90.96	2.93	1.55	10	1000
			OAO	88.70	3.37	10.9	10^4	1	2154	89.80	2.73	4.73	20	1000

Table 5
Comparison between the NN, SVM-OAA, SVM-OAO, ELM-OAA and ELM-OAO on database BioID.

BioID	NN		OAA	SVM						ELM				
(*P*-dim)	Testing rate (%)	Testing dev (%)	or OAO	Testing rate (%)	Testing dev (%)	Training time (s)	*C*	γ	# SVs	Testing rate (%)	Testing dev (%)	Training time (s)	*C*	*L*
PCA (5)	92.64	1.36	OAA	97.92	0.78	0.71	10^3	20	419	98.35	0.77	2.09	10	1000
(110)			OAO	95.73	1.20	4.09	50	5	162	95.62	1.24	1.51	0.2	1000
PCA (4)	90.81	1.88	OAA	96.64	1.46	0.53	10^4	20	384	96.58	1.66	1.54	50	1000
(88)			OAO	93.79	1.93	3.61	50	5	141	93.71	1.99	1.22	10^3	1000
PCA (3)	88.01	2.55	OAA	95.61	1.75	0.46	10^3	20	341	94.91	1.96	1.05	5	1000
(66)			OAO	91.35	2.62	3.21	10^3	20	116	91.43	2.56	0.97	0.2	1000
PCA (2)	81.81	2.73	OAA	90.74	2.49	0.37	10^4	20	284	90.17	1.66	0.64	0.2	1000
(44)			OAO	84.43	3.27	2.90	10^3	20	85	84.54	3.28	0.66	10^3	1000
LDA (5)	98.68	0.63	OAA	98.81	0.54	0.63	2	0.4	486	98.78	0.57	0.49	10^4	1000
(21)			OAO	98.69	0.5	3.38	20	2	93	98.67	0.55	0.87	0.2	1000
LDA (4)	98.48	0.77	OAA	98.52	0.86	0.52	5	1	400	98.44	0.92	0.40	5	1000
(21)			OAO	98.38	0.89	3.60	0.5	0.4	148	98.34	0.88	0.64	10	1000
LDA (3)	97.04	1.57	OAA	97.58	1.18	0.47	10	2	370	97.55	1.10	0.30	0.5	1000
(21)			OAO	97.34	1.11	3.34	0.5	0.4	132	97.22	1.05	0.59	50	1000
LDA (2)	88.34	2.60	OAA	90.91	1.84	0.37	50	1	342	90.63	2.04	0.20	0.5	1000
(21)			OAO	90.63	1.61	2.83	50	5	83	90.50	1.76	0.54	10^4	1000
DLA (5)	98.91	0.37	OAA	99.17	0.31	0.50	10^4	20	242	99.17	0.30	0.77	100	1000
			OAO	98.83	0.34	3.30	10^4	0.4	1755	98.90	0.45	2.16	100	1000
DLA (4)	98.21	1.12	OAA	98.72	0.88	0.45	50	2	252	98.92	0.85	0.66	5	1000
			OAO	98.24	1.10	3.23	10^4	20	800	98.24	1.13	1.98	2	1000
DLA (3)	96.93	1.19	OAA	97.65	1.25	0.46	10^3	20	266	97.78	1.20	0.53	50	1000
			OAO	96.95	1.30	3.04	10^4	0.4	1072	97.06	1.16	1.80	20	1000
DLA (2)	92.36	2.89	OAA	94.41	2.06	0.38	100	5	236	94.77	2.21	0.41	100	1000
			OAO	92.11	2.52	2.83	10^4	2	567	92.45	2.49	1.61	2	1000

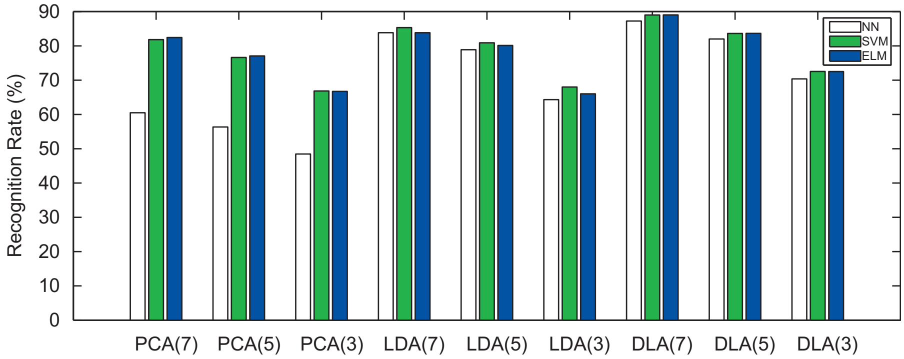

Fig. 5. Comparison of the generalization performance of NN, SVM-OAA and ELM-OAA on YALE.

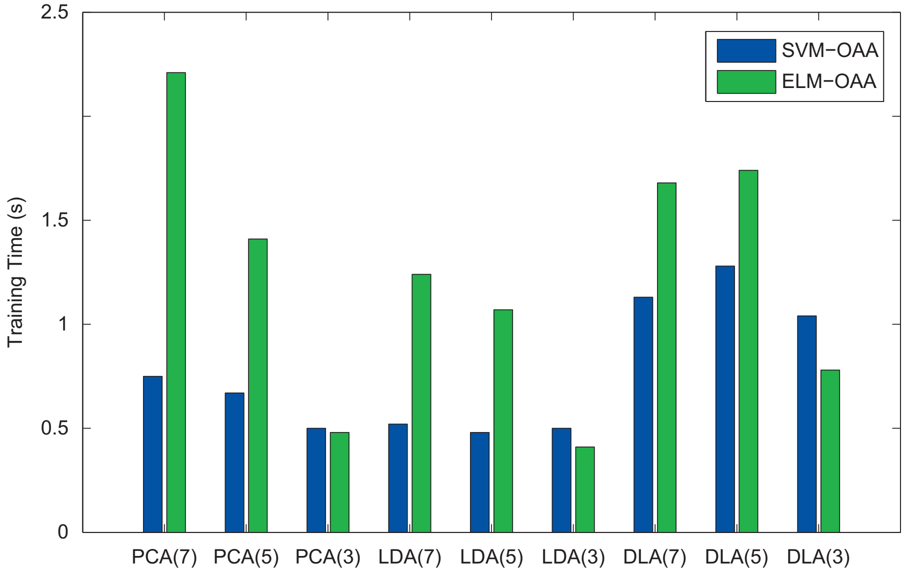

Fig. 6. Comparison of the training time of SVM-OAA and ELM-OAA on YALE.

well as feature-based learning method (AAM) on four popular face databases. The comparison has been made among ELM-OAA/ELM-OAO, nearest neighbor (NN), and SVM-OAA/SVM-OAO.

4.1. Face databases

In holistic learning the three databases to be tested are: YALE [33], ORL,[1] UMIST [34]. The YALE database consists of 15 individuals with 11 images for each individual. The images vary in illumination, facial expression, and decorations, e.g. when a person with or without glasses. The ORL database is collected from 40 persons, each one having 10 images with different illuminations, and expressions like smile, eye closing, etc. There are 564 facial images in the UMIST database from 20 individuals with different races. The participants are requested to take photos from profile to frontal view.

Different from the previous three databases, BioID[2] database has large variation on illumination, background, and even face sizes. With the annotation on facial images[3] and AAM [7], we obtain the feature representation about shape, appearance, and texture of each image. In the case of database BioID, feature vector with dimensionality 481 is obtained for each image.

For YALE database, the images are in the size of 40 × 40 pixels; for UMIST and ORL, the size of face images is 32 × 32 with 256 gray levels representing each pixel. And before training, the gray level of each image is normalized into the range of [0,1].

The sample images we use in holistic learning are displayed in Fig. 2. Sample images from BioID are displayed in Fig. 3.

4.2. Experimental results

For the YALE, ORL, UMIST face databases, we randomly choose 3, 5 and 7 images from each person for training and let the remaining for testing. For the BioID face database, 2, 3, 4 and 5 images are randomly selected from each person for training.

In both holistic and feature-based learning systems, PCA, LDA, and DLA are applied to transform the original data to the low-dimensional space, respectively. After transformed, different classifiers are tested. There is no parameter to tune in the NN classifier. As in [23], for SVM with RBF kernel: $K_{SVR}(\mathbf{x},\mathbf{x}_i)=\exp(-\gamma|\mathbf{x}-\mathbf{x}_i|^2)$, 225 different pairs of (C,γ) are tried which include 15 different values of C (0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 1000, 10 000) and 15 different values of γ (0.0001, 0.001, 0.01, 0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 20, 100, 1000, 10 000). While in ELM-OAA and ELM-OAO with sigmoid kernel: $K_{\text{ELM}}(\mathbf{x},\mathbf{x}_i)=\mathbf{h}(\mathbf{x})\cdot\mathbf{h}(\mathbf{x}_i)=\sum_{s=1}^{L}G(\mathbf{a}_s,b_s,\mathbf{x})G(\mathbf{a}_s,b_s,\mathbf{x}_i)$ where $G(\mathbf{a},b,\mathbf{x})=$

[1] http://www.uk.research.att.com/facedatabase.html

[2] http://www.bioid.com/support/downloads/software/bioid-face-database.html

[3] http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/

$1/(1+\exp(-(\mathbf{a}\cdot\mathbf{x}+b)))$, 15 different values of C (0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 1000, 10000) and 10 different values of L (20, 30, 50, 80, 100, 200, 500, 1000, 2000, 3000) have been tried, resulting in 150 different pairs in total.

Similar to [10], four parameters in DLA are tuned in order to reach the optimal recognition accuracy. They are: the projected dimensionality (P-dim), k_1 which represents the number of samples selected from the same class for a local patch, k_2 which

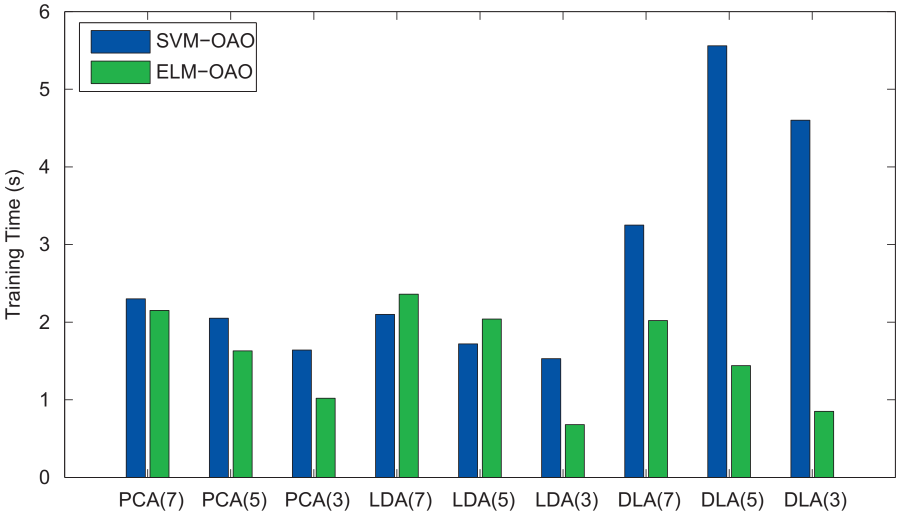

Fig. 7. Comparison of the training time of SVM-OAO and ELM-OAO on YALE.

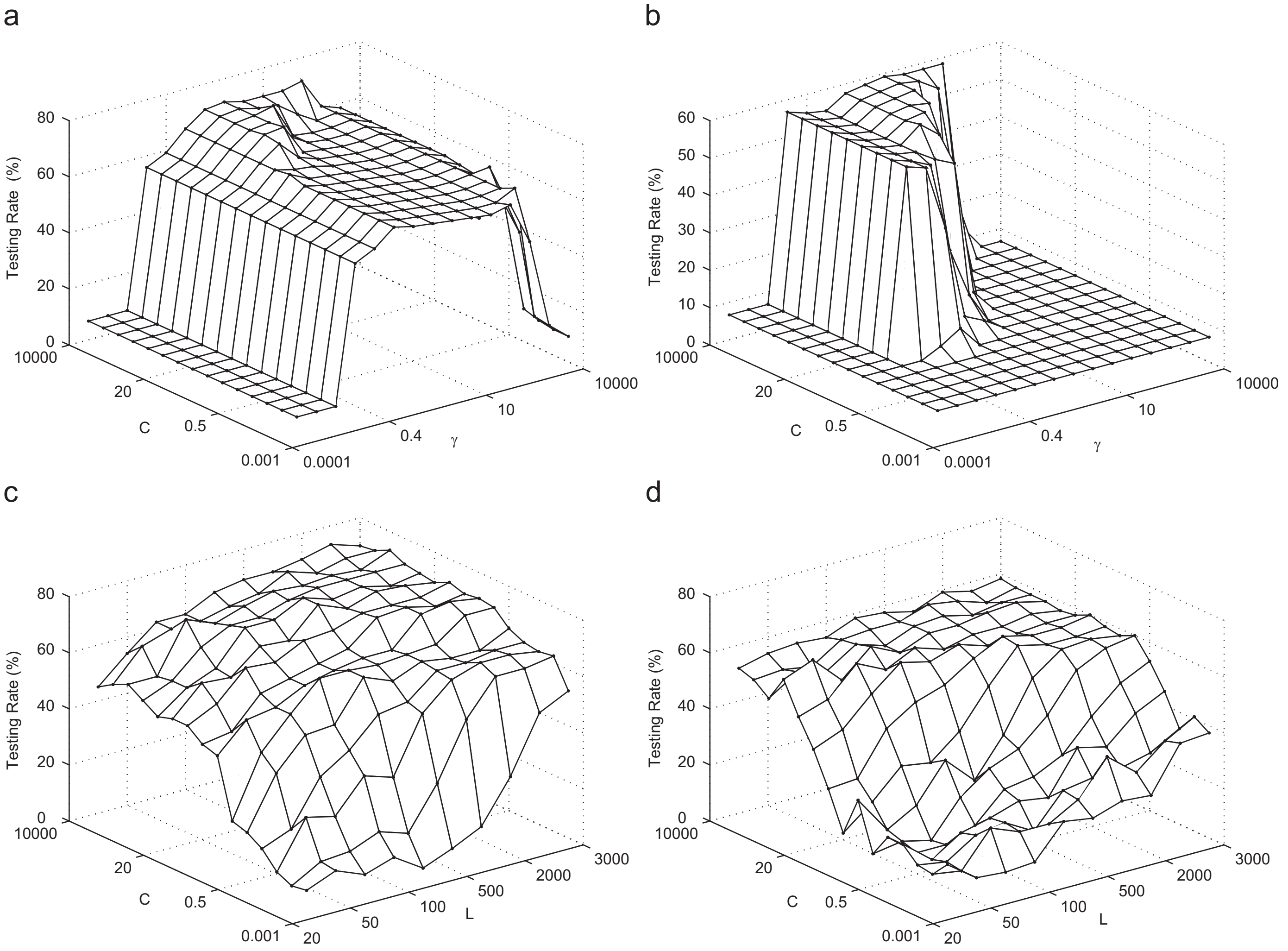

Fig. 8. The effect of tunable parameters ((C,γ) for SVM and (C,L) for ELM) on the generalization performance in holistic learning with three training samples from YALE. (a) PCA+SVM-OAA, (b) PCA+SVM-OAO, (c) PCA+ELM-OAA, and (d) PCA+ELM-OAO.

represents the number of samples selected from different classes for that local patch, and C_{DLA} which determines the proportion of within-class distance and between-class distance. The final choice of the parameters is listed in Table 1.

The average recognition result is obtained based on 20 trials for each case. In each trial, the training and testing data sets are randomly generated from the face images.

All the experiments are conducted in MATLAB 7.1 running in a PC with Pentium 4, 2.99 GHZ CPU.

Fig. 4 illustrates the two-dimensional representation of the images of five selected individuals from YALE, before and after PCA, LDA and DLA projection, respectively. The projection is obtained based on five training samples per person. Similar figures can be obtained for the cases with different number of training samples per person. Each point represents one image, and images from different identities are assigned with different shapes. The representation is obtained from the two leading coordinates of the PCA components and LDA/DLA projection applied on the original images. It can be found that after LDA projection, images from different individuals are well separated. But images from the same person are so close to each other and hard to distinguish. By revealing the inner structure of the data, the superposition of images from the same person is greatly reduced after DLA projection is applied. In all figures and tables, the number listed after each representation method shows how many images from each person are used for training. For example, in Fig. 4(b) LDA(5) means five images from each person for training in the LDA method.

The recognition performance for the five classifiers on four databases are shown from Tables 2–5. The number of support vectors for SVM-based classifier illustrates the size of the support vector network.

Fig. 5 represents the comparison of the recognition rate among the three classifiers using OAA. Similar figures can be obtained for the case of OAO. Figs. 6 and 7 illustrate the comparison of training time between SVM and ELM with OAA and OAO, respectively. Fig. 8 shows the relationship of the recognition rate and parameter selection for SVM-OAA and SVM-OAO trained on three sample images per person from YALE database after PCA projection. Similar figures can be obtained for the cases where LDA/DLA and different databases are tested.

4.3. Discussions

Seen from Tables 2–5, and Fig. 5, both SVM and ELM classifier outperform NN classifier, especially when learning method such as

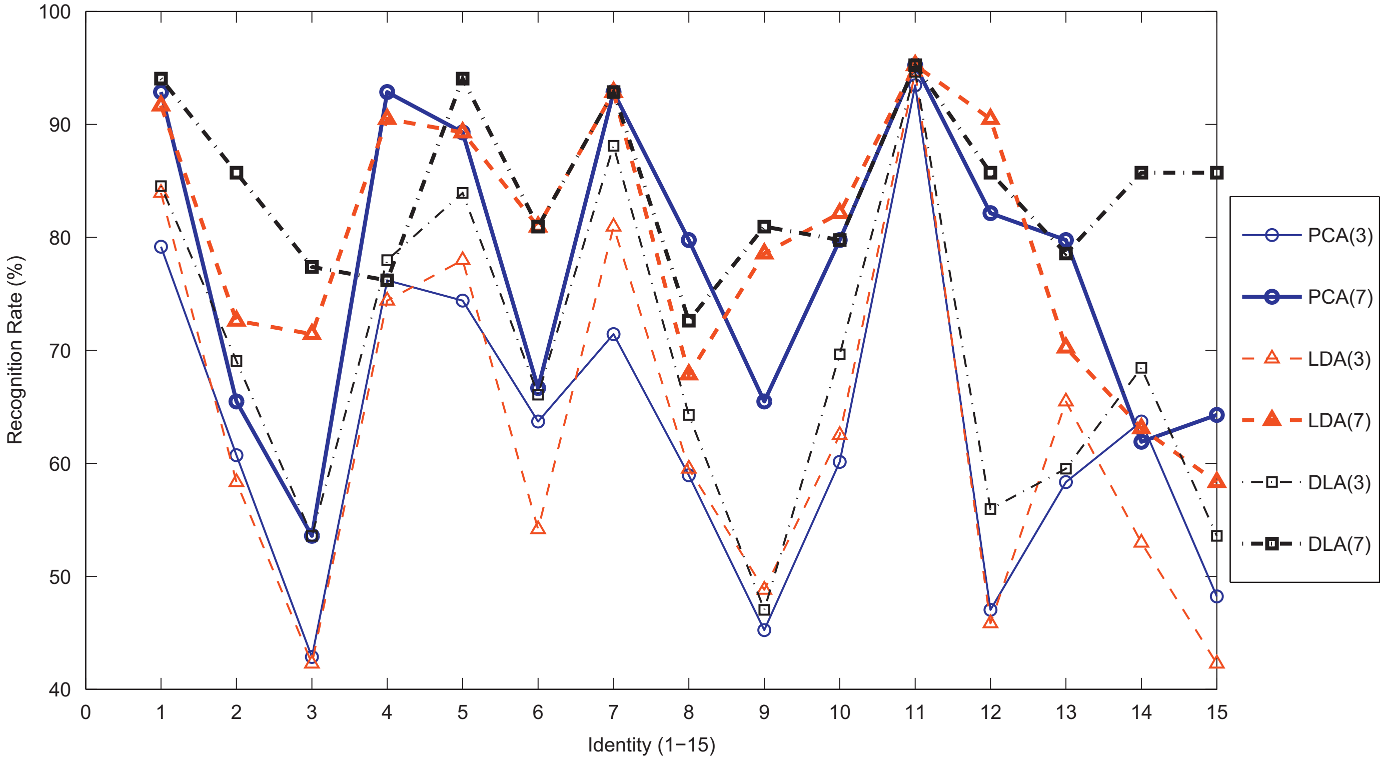

Fig. 9. The recognition rate for each person from database YALE using ELM-OAA.

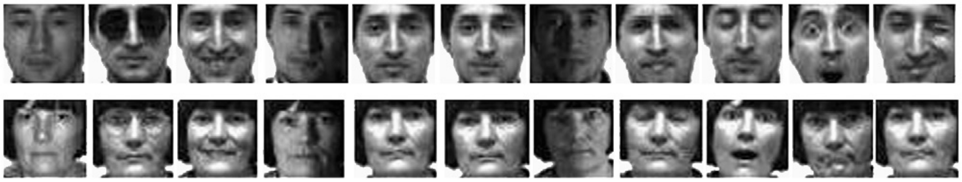

Fig. 10. Row 1: Images of Identity 3 from YALE; Row 2: Images of Identity 11 from YALE.

PCA is adopted or when the number of images used for training is quite limited. However, among the three dimensionality reduction methods, face recognition system with DLA usually achieves best performance.

The performance of our proposed ELM multi classifiers is found comparable to SVM in aspect of recognition accuracy and deviation. And YALE, with various facial expressions and illuminations, has the largest standard deviation among all the databases.

In our experiments, for most of the cases, OAA-based multi-label classifiers work better than the corresponding OAO-based multi-label classifiers. The training time of ELM is closely related to the number of hidden nodes. When the number of hidden nodes is around 1000, the training time of ELM is quite close to SVM.

In our experiment L is fixed as 1000. It is found from the experimental result that better performance can be obtained when the number of hidden nodes is attentively tuned.

Seen from Fig. 8, we can find that ELM is not sensitive to the parameters. The good result is achieved normally when L is large enough. For SVM the testing rate is very sensitive to the choice of γ, especially when OAO is adopted. That is to say, there are two parameters to tune one after another. Therefore, it is much easier to use ELM since we only need to tune one parameter.

In addition, the recognition rate of ELM-OAA with respect to each person from database YALE is shown in Fig. 9, where the six plot lines indicate different number of training samples (e.g, 3 and 7 samples each person. Similar figure can be obtained for the case where five samples are used from each person), and different dimensionality reduction methods used (PCA, LDA or DLA). Similar figure can be obtained for the SVM case. It is obvious that some identities (Identity 11 for example) tend to have better recognition rate and are robust to different configurations of the recognition system. On the other hand, some identities (Identity 3 for example) are difficult to be recognized. However, it is shown that with proper choice of configuration, the recognition rate can be greatly improved. The possible reason of this tendency of each individual can be seen from the displayed samples of Identity 3 and 11 (Fig. 10). The images of Identity 3 have relatively large variations in illumination and facial expression. Different from Identity 3, images of Identity 11 have much less variations in those aspects.

5. Conclusions

In this paper, we have proposed ELM-OAA and ELM-OAO and tested them in face recognition applications. The performance of ELM-OAA and ELM-OAO is compared with SVM-OAA and SVM-OAO in both holistic learning and feature-based learning environments, with different dimensionality reduction methods. With comparable recognition accuracy and training time, ELM for classification is better in the convenience for parameter selection. In ELM for classification, only one parameter (C) needs to be varied after another one (L) is assigned with a large value.

References

[1] S.-H. Lin, S.-Y. Kung, L.-J. Lin, Face recognition/detection by probabilistic decision-based neural network, IEEE Transactions on Neural Networks 8 (1) (1997) 114–132.

[2] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Computing Surveys 35 (4) (2003) 399–458, doi:10.1145/954339.954342.

[3] G. Guo, S.Z. Li, K. Chan, Face recognition by support vector machines, in: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, March 26–30, 2000, pp. 196–201.

[4] B. Heisele, P. Ho, J. Wu, T. Poggio, Face recognition: component-based versus global approaches, Computer Vision and Image Understanding 91 (1–2) (2003) 6–21.

[5] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[6] W. Zhao, A. Krishnaswamy, R. Chellappa, Discriminant analysis of principal components for face recognition, in: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, April 14–16, 1998, pp. 336–341.

[7] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (1998) 681–685.

[8] Z. Li, X. Tang, Using support vector machines to enhance the performance of Bayesian face recognition, IEEE Transactions on Information Forensics and Security 2 (2) (2007) 174–180.

[9] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.

[10] T. Zhang, D. Tao, J. Yang, Discriminative locality alignment, in: Proceedings of the Tenth European Conference on Computer Vision (ECCV 08), Marseille, France, October 12–18, 2008, pp. 725–738.

[11] P.J. Phillips, Support vector machines applied to face recognition, in: Proceedings of Advances in Neural Information Processing Systems II, Denver, Colorado, USA, 1999, pp. 803–809.

[12] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, The Journal of Machine Learning Research 1 (2001) 113–141.

[13] G.-Y. Zhang, S.-Y. Peng, H.-M. Li, Combination of dual-tree complex wavelet and SVM for face recognition, in: Proceedings of International Conference on Machine Learning and Cybernetics, vol. 5, Kunming, China, July 12–15, 2008, pp. 2815–2819.

[14] J.-Y. Gan, S.-B. He, Face recognition based on 2DLDA and support vector machine, in: Proceedings of International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR 2009), Baoding, China, July 12–15, 2009, pp. 211–214.

[15] L. Zhao, Y. Song, Y. Zhu, C. Zhang, Y. Zheng, Face recognition based on multi-class svm, in: Proceedings of Chinese Control and Decision Conference (CCDC 09), Guilin, China, June 17–19, 2009, pp. 5871–5873.

[16] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.

[17] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of International Joint Conference on Neural Networks (IJCNN 2004), vol. 2, Budapest, Hungary, July 25–29, 2004, pp. 985–990.

[18] H.-J. Rong, G.-B. Huang, Y.-S. Ong, Extreme learning machine for multi-categories classification applications, in: Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2008) (IEEE World Congress on Computational Intelligence), Hong Kong, June 1–8, 2008, pp. 1709–1713.

[19] Y. Lan, Y.C. Soh, G.-B. Huang, Extreme learning machine-based bacterial protein subcellular localization prediction, in: Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2008) (IEEE World Congress on Computational Intelligence), Hong Kong, June 1–8, 2008, pp. 1859–1863.

[20] T. Helmy, Z. Rasheed, Multi-category bioinformatics dataset classification using extreme learning machine, in: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation (CEC 09), Trondheim, Norway, May 18–21, 2009, pp. 3234–3240.

[21] C.-W.T. Yeu, M.-H. Lim, G.-B. Huang, A. Agarwal, Y.-S. Ong, A new machine learning paradigm for terrain reconstruction, IEEE Geoscience and Remote Sensing Letters 3 (July) (2006) 382–386.

[22] J. Lu, Y. Zhao, Y. Xue, J. Hu, Palmprint recognition via locality preserving projections and extreme learning machine neural network, in: Proceedings of Ninth International Conference on Signal Processing (ICSP 2008), Beijing, China, October 26–29, 2008, pp. 2096–2099.

[23] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.

[24] K. Fukunag, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1990.

[25] D. Cai, X. He, J. Han, SRDA: An efficient algorithm for large-scale discriminant analysis, IEEE Transactions on Knowledge and Data Engineering 20 (January) (2008) 1–12.

[26] C. Cortes, V.N. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.

[27] R. Fletcher, Practical Methods of Optimization: Volume 2 Constrained Optimization, Wiley, New York, 1981.

[28] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks 17 (4) (2006) 879–892.

[29] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.

[30] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.

[31] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[32] B. Kijsirikul, N. Ussivakul, Multiclass support vector machines using adaptive directed acyclic graph, in: Proceedings of International Joint Conference on Neural Networks (IJCNN 02), Honolulu, Hawaii, USA, May 12–17, 2002, pp. 980–985.

[33] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[34] D.B. Graham, N.M. Allinson, Characterizing virtual eigensignatures for general purpose face recognition, in: H. Wechsler, J.P. Phillips, V. Bruce, F.F. Soulie, T.S. Huang (Eds.), Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and System Sciences, vol. 163, Springer, Berlin, 1998.

Weiwei Zong received the B.E. degree from Central South University, China in 2008. She is currently a Ph.D. student with school of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her research interests include neural networks, support vector machines and pattern recognition.

Guang-Bin Huang received the B.Sc degree in applied mathematics and M.E. degree in computer engineering from Northeastern University, P. R. China, in 1991 and 1994, respectively, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 1999. During undergraduate period, he also concurrently studied in Applied Mathematics department and Wireless Communication department of Northeastern University, P. R. China.

From June 1998 to May 2001, he worked as Research Fellow in Singapore Institute of Manufacturing Technology (formerly known as Gintic Institute of Manufacturing Technology) where he has led/implemented several key industrial projects. From May 2001, he has been working as an Assistant Professor and Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University. His current research interests include machine learning, computational intelligence, and extreme learning machine. He serves as an Associate Editor of Neurocomputing and IEEE Transactions on Systems, Man and Cybernetics—Part B. He is a senior member of IEEE.