J Sign Process Syst (2013) 73:73–81
DOI 10.1007/s11265-013-0730-x

Evolutionary Extreme Learning Machine and Its Application to Image Analysis

Nan Liu · Han Wang

Received: 3 September 2010 / Revised: 28 October 2012 / Accepted: 1 January 2013 / Published online: 26 January 2013

Abstract Extreme learning machine (ELM) and evolutionary ELM (E-ELM) were proposed as a new class of learning algorithm for single-hidden layer feedforward neural network (SLFN). In order to achieve good generalization performance, E-ELM calculates the error on a subset of testing data for parameter optimization. Since E-ELM employs extra data for validation to avoid the overfitting problem, more samples are needed for model training. In this paper, the cross-validation strategy is proposed to be embedded into the training phase so as to solve the overtraining problem. Based on this new learning structure, two extensions of E-ELM are introduced. Experimental results demonstrate that the proposed algorithms are efficient for image analysis.

Keywords Extreme learning machine · Differential evolution · Image analysis · Face recognition

1 Introduction

With rapid advancement of computer and database technologies, understanding and mining useful information from huge amount of data attract numerous efforts from the areas of databases, machine learning, and statistics [12]. Pattern recognition is the study of how computers sense the environment, learn from stored patterns of interest, and make decisions to categorize unseen data. Recognizing patterns is an easy task to human, whereas it is difficult for machines to accomplish. Nevertheless, since computers have several advantages on processing speed and data storage compared with human, many pattern recognition techniques have been proposed and applied to a variety of scientific disciplines including computer vision, image understanding, speech recognition, computational biology and so on. Image analysis [16] is one of the most studied problems in pattern recognition, which has been widely used in many applications such as face detection and recognition.

To accomplish the task of recognition, choosing a suitable classifier plays an important role in both the training and testing phases. During the learning stage, the classification rule is formed by collecting knowledge from training samples, then the well established classifier is applied to categorize unseen testing data. In supervised learning, classifiers always suffer from overtraining which may degrade the generalization performance. In other words, although small training errors are obtained in the training phase, the testing result might be unsatisfactory. It is observed that the sets of patterns misclassified by different classifiers would not necessarily overlap which suggests that combining the outputs of various classifiers has potential to offer better prediction results. Therefore, the ensemble-based decision making strategy [18] is possible to be adopted for constructing reliable image analysis systems. Moreover, several techniques have recently been proposed to improve the generalization performance of the learning system by either maximizing the uncertainty [23, 26] or combining multiple reducts of rough sets [24].

Extreme learning machine (ELM) was proposed recently as an efficient learning algorithm for single-hidden layer feedforward neural network (SLFN) [10, 11]. It increases the learning speed by randomly generating weights and

N. Liu (✉)
Department of Emergency Medicine, Singapore General Hospital, Outram Road, Singapore 169608, Singapore
e-mail: nliu@pmail.ntu.edu.sg

H. Wang
School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
e-mail: hw@ntu.edu.sg

biases for hidden nodes rather than iteratively adjusting network parameters that is commonly adopted by gradient-based neural networks (NN). Although ELM is fast and presents good generalization performance, there are still a lot of room for further improvements. Zhu et al. [28] claimed that random assignment of parameters will introduce un-optimal input weights and hidden biases. Consequently, evolutionary extreme learning machine (E-ELM) was proposed by taking advantages of both ELM and differential evolution (DE) [22] to remove redundancy among hidden nodes and achieve satisfactory performance with more compact networks. Furthermore, pruned-ELM (P-ELM) was presented by Rong et al. [19]. Their idea is to initialize a large network and prune it during learning. Apart from numerous improvements [8, 9], ELM was also implemented in microarray data classification [27] and showed its superiority to support vector machines.

In this paper, we propose using the cross-validation strategy for E-ELM training to solve the classification problem. Classifiers usually suffer from overtraining in supervised learning, which might degrade the generalization performance. During the training phase, training samples are categorized into several classes by classifier and the learning error is used to evaluate the efficiency of training. In general, minimum training error is expected, but it cannot guarantee good recognition results on unseen data. The main mechanism behind our proposal is partitioning the original training set using cross-validation scheme into R subsets and then R pairs of training and validation sets are obtained so that each training set consists of $(R-1)$ subsets. In the new training procedure, each of the R learners is constructed using $(R-1)$ subsets and validated with the remaining single subset. The cross-validation process is then repeated R times, with each of the R subset used exactly once for validation. Subsequently, in the extensions of E-ELM, the averaged classification accuracy (CA) across all R trials is employed as the fitness function for selecting the most fitting network parameters for testing. The above mentioned learning procedure is reasonable to avoid overfitting because the validation set (N/R training samples) is used to replace the entire training set to evaluate the learning error in each one of the R classifiers. As a result, cross-validation based E-ELM (E-ELMcv) and cross-validation based improved E-ELM (IE-ELMcv) are proposed.

2 Preliminaries

2.1 Extreme Learning Machine

As one of learning algorithms for SLFN, ELM randomly selects weights and biases for hidden nodes, and analytically determines the output weights by finding least square solution. Given a training set consisting of N samples $L = \{(\mathbf{x}_j, \mathbf{t}_j)|\mathbf{x}_j \in \mathbf{R}^n, \mathbf{t}_j \in \mathbf{R}^m, j = 1, 2, ..., N\}$, where $\mathbf{x}_j$ is an $n \times 1$ input vector and $\mathbf{t}_j$ is an $m \times 1$ target vector, an SLFN with $\tilde{N}$ hidden nodes is formulated as

$$f_{\tilde{N}}(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, 2, ..., N \quad (1)$$

where additive hidden node is employed. $\mathbf{w}_i$ is n-dimensional weight vector connecting ith hidden node and input neurons. In approximating N samples using $\tilde{N}$ hidden nodes, $\boldsymbol{\beta}_i$, $\mathbf{w}_i$, and b_i are supposed to exist if zero error is obtained. Consequently, Eq. 1 can be written in a more compact format as $\mathbf{H}\hat{\boldsymbol{\beta}} = \mathbf{T}$ where $\mathbf{H}(\mathbf{w}_1, ..., \mathbf{w}_{\tilde{N}}, b_1, ..., b_{\tilde{N}}, \mathbf{x}_1, ..., \mathbf{x}_N)$ is hidden layer output matrix of the network, $h_{ji} = g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)$ is the output of ith hidden neuron with respect to $\mathbf{x}_j$, $i = 1, 2, ..., \tilde{N}$ and $j = 1, 2, ..., N$; $\hat{\boldsymbol{\beta}} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{\tilde{N}}]^{\mathrm{T}}$ and $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_N]^{\mathrm{T}}$ are the output weight matrix and the target matrix, respectively.

$$\mathbf{H}(\mathbf{w}_1, ..., \mathbf{w}_{\tilde{N}}, b_1, ..., b_{\tilde{N}}, \mathbf{x}_1, ..., \mathbf{x}_N) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (2)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^{\mathrm{T}} \end{bmatrix}_{\tilde{N} \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{t}_N^{\mathrm{T}} \end{bmatrix}_{N \times m} \quad (3)$$

Huang et al. [11] pointed out that in real applications training error cannot be made exactly zero as the number of hidden nodes $\tilde{N}$ will always be less than the number of training samples N. To obtain small non-zero training error, Huang et al. [11] proposed randomly assigning values for parameters $\mathbf{w}_i$ and b_i, and thus the system becomes linear so that the output weights can be estimated as $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$, where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse [21] of the hidden layer output matrix $\mathbf{H}$. Given a training set L_{trn}, activation function $g(x)$, and hidden node number $\tilde{N}$, the ELM algorithm can be summarized as follows.

1. Generate parameters $\mathbf{w}_i$ and b_i for $i = 1, ..., \tilde{N}$,
2. Calculate the hidden layer output matrix $\mathbf{H}$,
3. Calculate the output weight using $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

2.2 Evolutionary Extreme Learning Machine

To eliminate possible non-optimum within hidden nodes and create more compact networks, an evolutionary extreme learning machine algorithm was introduced [28]. E-ELM deploys DE to select optimal weights and biases. At the beginning, E-ELM initializes a population of N_p parameter vectors $\{\mathbf{z}_{p,G}|p = 1, 2, ..., N_p\}$, and then chooses the

best individual in terms of fitness to form a new generation in which the selection pool contains candidates from Gth generation and their variants after operations of mutation and crossover.

$$E = \sqrt{\frac{\sum_{j=1}^{N} \| \sum_{1}^{\tilde{N}} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_j) - \mathbf{t}_j \|^2}{m \times N}} \tag{4}$$

In E-ELM, the fitness of each individual is defined as root mean squared error (RMSE) shown as Eq. 4 on validation set instead of whole training set [28]. In addition, the norm of output weights $\|\boldsymbol{\beta}\|$ is considered as another criterion to improve the generalization performance.

3 Proposed Methods

E-ELM [28] is one of the successful improvements on ELM. In this section, we propose two E-ELM based extensions to enhance the ability of classification.

3.1 Cross-validation Based Evolutionary Extreme Learning Machine (E-ELMcv)

E-ELM was proposed by employing both RMSE and $\boldsymbol{\beta}$ on the validation set for candidate selection to achieve better classification accuracy with more compact networks. Since the original testing set needs to be evenly separated into testing set and validation set, E-ELM uses extra data for training which might not be suitable for applications in which testing samples are limited. Therefore, the validation set is crucial to E-ELM learning. Alternatively, samples from training set can be used to form the validation set, but the number of training samples decreases and that could affect the generalization performance. Hence, we propose the E-ELMcv algorithm to avoid using an extra validation set for training.

In order to inherit the merit of E-ELM, the proposed algorithm also deploys differential evolution (DE) as a tool to select optimal weights and biases for hidden nodes. At first, a set of parameter vectors $\{\mathbf{z}_{p,G} | p = 1, 2, ..., N_p\}$ is initialized, in which components of $\mathbf{z}_{p,G}$ have bound as $[-1, 1]$.

$$\mathbf{z}_{p,G} = [w_{11}, ..., w_{1\tilde{N}}, ..., w_{n1}, ..., w_{n\tilde{N}}, b_1, ..., b_{\tilde{N}}] \tag{5}$$

where G denotes Gth generation, and the input weights $\mathbf{w}_i$ and hidden node biases b_i form the candidate vector. The size of vector depends on the number of hidden nodes $\tilde{N}$ and feature dimension n. DE updates the population under the driven of fitness function. Before creating a new generation, mutation, crossover, and selection operations are applied. In details, for each vector $\mathbf{z}_{p,G}$, a mutant vector is generated according to

$$\hat{\mathbf{z}}_{p,G+1} = \mathbf{z}_{r_1,G} + F \cdot (\mathbf{z}_{r_2,G} - \mathbf{z}_{r_3,G}) \tag{6}$$

where $r_1, r_2, r_3 \in \{1, 2, ..., N_p\}$ are the random indices and F is a positive real number not larger than 2, which is a factor to control amplification of differential variation $(\mathbf{z}_{r_2,G} - \mathbf{z}_{r_3,G})$. Subsequently the crossover operator is introduced to increase diversity among population. As a result, the D-dimensional vector is constructed as

$$\tilde{\mathbf{z}}_{p,G+1} = (\tilde{z}_{1p,G+1}, \tilde{z}_{2p,G+1}, ..., \tilde{z}_{Dp,G+1}) \tag{7}$$

where $D = \tilde{N}(n + 1)$, and we have

$$\tilde{z}_{qp,G+1} = \begin{cases} \hat{z}_{qp,G+1} & \text{randb}(q) \le \text{CR} \\ & \text{or } q = \text{rnbr}(p) \\ z_{qp,G} & \text{randb}(q) > \text{CR} \\ & \text{and } q \ne \text{rnbr}(p) \end{cases} \tag{8}$$

In Eq. 8, $q \in \{1, 2, ..., D\}$, and the qth evaluation of a uniform random number generator with outcome in $[0, 1]$ is determined by randb(q). CR is a user-defined constant in $[0, 1]$. A random index rnbr(p) is used to ensure that at least one parameter from $\hat{\mathbf{z}}_{p,G+1}$ is obtained by $\tilde{\mathbf{z}}_{p,G+1}$.

Prior to selection, fitness values are calculated for all $\mathbf{z}_{p,G}$ and $\tilde{\mathbf{z}}_{p,G+1}$ where $p = 1, 2, ..., N_p$. The fitness function plays a key role in candidate selection. We apply classification accuracy (CA) as the sole component in the fitness compared to the combinatorial usage of RMSE and $\|\boldsymbol{\beta}\|$ in E-ELM. We choose a new fitness function primarily due to two reasons: First, the aim of the proposed method is for the purpose of classification rather than regression, therefore a fitness function based on prediction accuracy is more straightforward than a fitness function based on RMSE; Second, the introduction of cross-validation strategy in E-ELMcv makes it difficult to implement $\|\boldsymbol{\beta}\|$ based selection as we have multiple values of $\|\boldsymbol{\beta}\|$. Then, by partitioning the training set L into R pairs of data sets using the cross-validation strategy, the fitness value of $\mathbf{z}_{p,G}$ can be evaluated as

$$\text{CA}^{p,G} = \frac{1}{R} \sum_{r=1}^{R} \text{CA}_r^{p,G} \tag{9}$$

and the fitness value for $\tilde{\mathbf{z}}_{p,G+1}$ is calculated as

$$\text{CA}^{p,G+1} = \frac{1}{R} \sum_{r=1}^{R} \text{CA}_r^{p,G+1} \tag{10}$$

If $\tilde{\mathbf{z}}_{p,G+1}$, the evolved candidate, appears fitter than the original parameter vector, i.e., $\text{CA}^{p,G+1} > \text{CA}^{p,G}$, $\tilde{\mathbf{z}}_{p,G+1}$ will

Figure 1 The architecture of the proposed E-ELMcv algorithm.

Algorithm: E-ELMcv

Inputs

- A set of N samples $L = \{(\mathbf{x}_j, \mathbf{t}_j) | \mathbf{x}_j \in \mathbf{R}^n, \mathbf{t}_j \in \mathbf{R}^m, j = 1, 2, ..., N\}$.
- Number of generation N_G, population size N_p, number of hidden nodes $\tilde{N}$, fold number R.

Initialization

- Randomly generate N_p candidate vectors $\mathbf{z}_{p,G} = [w_{11}, ..., w_{1\tilde{N}}, ..., w_{n1}, ..., w_{n\tilde{N}}, b_1, ..., b_{\tilde{N}}]$ to comprise a population of possible solutions.
- Partition the training set into R groups using R-fold cross-validation. In each group, $(R-1)N/R$ samples are used for training and the rest of N/R samples form the validation set.

For $k = 1, 2, ..., N_G$

1. Mutation: $\hat{\mathbf{z}}_{p,G+1} = \mathbf{z}_{r_1,G} + F \cdot (\mathbf{z}_{r_2,G} - \mathbf{z}_{r_3,G})$ with random indices $r_1, r_2, r_3 \in \{1, 2, ..., N_p\}$.
2. Crossover: $\tilde{\mathbf{z}}_{p,G+1} = (\tilde{z}_{1p,G+1}, \tilde{z}_{2p,G+1}, ..., \tilde{z}_{Dp,G+1})$ where $D = \tilde{N}(n+1)$.
3. Selection: Evaluate the fitness values for the original candidate $\mathbf{z}_{p,G}$ and its evolved vector $\tilde{\mathbf{z}}_{p,G+1}$ as $(1/R)\sum_{r=1}^{R} \mathrm{CA}_r^{p,G}$ and $(1/R)\sum_{r=1}^{R} \mathrm{CA}_r^{p,G+1}$, and select the fitter one as $\mathbf{z}_{p,G+1}$.
4. Repeat the above three steps until the number of iterations has achieved.

End for

Prediction

- Obtain the best candidate vector $\mathbf{z}^{\mathrm{best}}$ in terms of achieving highest cross-validation accuracy.
- Predict labels for unknown patterns with $\mathbf{w}_i$ and b_i in $\mathbf{z}^{\mathrm{best}}$ as the parameters for E-ELM.

be selected into the next generation instead of $\mathbf{z}_{p,G}$; otherwise, $\mathbf{z}_{p,G}$ is considered as the elite candidate and continues to survive in $(G+1)$th generation as $\mathbf{z}_{p,G+1}$. After a number of iterations, the best candidate vector $\mathbf{z}^{\mathrm{best}}$ in terms of achieving highest prediction accuracy is obtained for testing. Given new patterns, predictions are carried out using $\mathbf{w}_i$ and b_i in $\mathbf{z}^{\mathrm{best}}$. Figure 1 illustrates the architecture of the proposed E-ELMcv algorithm.

3.2 Cross-validation Based Improved Evolutionary Extreme Learning Machine (IE-ELMcv)

It is not surprising that only partial hidden nodes contribute to classification positively. In other words, redundancy exists in hidden layer which may weaken the generalization performance. Rong et al. [19] proposed P-ELM algorithm to initialize a large network and prune it by removing irrelevant hidden nodes during training.

Both IE-ELMcv and E-ELMcv share the same architecture described in Fig. 1 except for several minor changes as the improvement. In IE-ELMcv algorithm, instead of deleting hidden nodes adaptively, we propose assigning constant values to some hidden nodes' $\mathbf{w}_i$ and b_i during training phase to control the contributions of certain nodes, i.e., the parameters of selected nodes are pre-defined "invariant" values but not randomly generated measures. The selection of N_u "invariant" nodes are determined by a random number in parameter vector in DE. Then the parameter vector becomes

$$\mathbf{z}_{i,G} = [w_{11}, ..., w_{1\tilde{N}}, ..., w_{n1}, ..., w_{n\tilde{N}}, b_1, ..., b_{\tilde{N}}, u] \quad (11)$$

where u is the factor from which the number of "invariant" hidden nodes are computed. The number N_u is estimated as

$$N_u = \left\lceil \frac{(u^2 + e_1)\tilde{N}}{e_2} \right\rceil, \quad u \in [-1, 1] \quad (12)$$

where $\lceil \cdot \rceil$ is a ceiling operator, and e_1 and e_2 are constants for limiting the ranges of N_u. For instance, if e_1 and e_2 are set to 0.1 and 5, N_u will be bounded between $\lceil 0.02 \times \tilde{N} \rceil$ and $\lceil 0.22 \times \tilde{N} \rceil$. Subsequently, N_u hidden nodes are randomly selected and the corresponding $\mathbf{w}_i$ and b_i are set to a constant value (it is set to 0.1 in this paper) such that non-optimum within input weights and hidden biases might be removed and the generalization performance could be improved. Though the network architecture keeps unchanged, the complexity of hidden layer has been reduced as the number of tunable variables ($\mathbf{w}_i$ and b_i) is decreased.

4 Performance Evaluation

Evaluations are carried out on four face databases with ELM, E-ELM, and our proposed algorithms for image

Table 1 Data sets used in the experiments.

Data set	Training	Testing	Dimension	Class
Combo	555	575	81	75
FERET	1280	1433	81	320
GTFD	400	350	81	50
ORL	200	200	81	40

analysis. All of the computerized simulations are run in MATLAB 7 environment under workstation equipped with Intel Pentium 4 3.2GHz CPU and 1G RAM. The learning and testing processes are repeated 50 times with sigmoid function $g(x) = 1/(1 + e^{-\lambda x})$ as the activation function. In this paper, 10-fold cross-validation is applied for training. In E-ELM and its variants, N_p, F, and CR are 50, 1, and 0.8, respectively. Furthermore, 0.1 and 5 are chosen as the values for e_1 and e_2. The number of generations is heuristically determined as 20. The data sets used in the experiments are summarized in the following section. Except for E-ELM, all approaches are trained with the entire training set. E-ELM divides testing data into two groups equally, and chooses one group as the validation set to avoid overtraining.

4.1 Databases

In assessing the performance, four sets of face images are employed (Table 1). They are FERET face database [17], ORL database [20], a combo face database (ORL, UMIST [6], and Yale [1]), and Georgia Tech face database (GTFD) [2]. Since the combo data set encompasses ORL, UMIST, and Yale database, there are five stand-alone image sets. The FERET database is a standard testing set for performance evaluation, including 14126 images from 1199 individuals with views ranging from frontal to left and right profiles. We adopt a pre-processed subset composed of 2713 face images from 320 subjects with each subject having at least six images with at most 45° of pose variation, which was used in Lu et al. [13]. Face images from the subset of the FERET database are manually aligned, cropped, and normalized to 32 × 32 pixels, with 256 gray levels per pixel. The ORL database contains 400 images of 40 individuals and half of these images are used for training and the rest for testing. The combo set consists of 555 training samples and 575 testing images in total, and all images belong to 75 different classes with large variations of illumination, poses, and facial expressions. In the Georgia Tech face database, each of 50 subjects has 15 images. All the color images

Figure 2 Examples of five stand-alone face databases used in the experiments: **a** GTFD, **b** ORL, **c** UMIST, **d** Yale and **e** FERET.

Table 2 The experimental results on four face databases where hidden nodes are set as 100 for all algorithms.

Data set	Classification algorithm	Training time (s)	Testing accuracy (%)
Combo	ELM	0.0672	81.82 ± 1.35
	E-ELM	88.641	81.90 ± 1.23
	E-ELMcv	160.59	84.02 ± 1.19
	IE-ELMcv	152.22	85.36 ± 1.28
	BP	570.35	78.85 ± 1.63
FERET	ELM	0.1069	41.65 ± 0.98
	E-ELM	152.02	41.91 ± 1.52
	E-ELMcv	308.49	43.72 ± 0.80
	IE-ELMcv	300.48	44.78 ± 0.87
	BP	1361.5	36.92 ± 1.93
GTFD	ELM	0.0555	54.31 ± 1.88
	E-ELM	81.141	57.17 ± 2.82
	E-ELMcv	125.58	60.29 ± 2.32
	IE-ELMcv	128.35	61.03 ± 2.28
	BP	516.21	46.67 ± 3.06
ORL	ELM	0.0391	78.45 ± 2.72
	E-ELM	62.813	80.70 ± 3.48
	E-ELMcv	93.588	82.67 ± 1.76
	IE-ELMcv	90.122	83.25 ± 2.15
	BP	357.79	71.05 ± 3.52

with cluttered background are taken at resolution 640 × 480 pixels where frontal and/or tilted faces with different facial expressions, lighting conditions and scale are presented. In the experiments, a pre-processed set of images with the background removed is adopted, and for each subject, eight samples are randomly selected for training and the rest of seven images are used for testing. Before the experimental evaluation, images in Yale and GTFD databases are manually cropped and resized to 112 × 92 to make their dimensions identical to those of samples in ORL and UMIST. Figure 2 presents examples after pre-processing from the above mentioned face databases. Furthermore, we apply the discrete cosine transform (DCT) [3, 7] to convert 2D face images to low-dimensional vectors of DCT coefficients so as to alleviate the computational burden for classification.

4.2 Experimental Results

The experimental results are presented in Table 2. It is shown that ELM is the fastest learner but receives poor performance in classification. Our proposed E-ELMcv and IE-ELMcv outperform ELM, E-ELM, and the backpropagation (BP) neural network [5] in terms of achieving higher testing accuracies on all data sets. In summary, the proposed methods are stable and efficient as they can provide good generalization performance. Before recording results for the extensions of E-ELM, several trials have been done and the testing outcomes indicate that E-ELMcv and IE-ELMcv need more training time than E-ELM. We reduce the population size N_p and the number of generations to half of their values, and discover that the learning time decreases dramatically. The results using the above new parameters for E-ELMcv and IE-ELMcv in Table 2 show that although the population size is shrunk and the evolving procedure is shortened, the proposed E-ELM based extensions can still achieve higher testing accuracies than E-ELM in comparable learning time. Moreover, it is observed that conventional gradient-based BP costs much longer time for training while its classification results are far from satisfactory.

To compare with state-of-the-art face recognition techniques such as Bayes method [15], linear discriminant analysis (LDA) [1], uncorrelated LDA (ULDA) [25], regularized version of revised direct LDA (R-JD-LDA) [14], we conducted several experiments on the FERET database and showed the comparison results in Table 3. There were three subsets of FERET face database used in the experiments, namely $C160$, $C240$ and $C320$ where the number of subjects were 160, 240 and 320, respectively. The original E-ELMcv and IE-ELMcv methods performed much better than LDA and ULDA on databases that contain more subjects. In general, E-ELMcv and IE-ELMcv cannot outperform Bayes and R-JD-LDA methods. However, it is worth noting that our proposed methods are focused on the aspect

Table 3 The comparison results between the proposed methods and four classical face recognition techniques on the FERET database.

C	Bayes	LDA	ULDA	R-JD-LDA	E-ELMcv	E-ELMcv[a]	IE-ELMcv	IE-ELMcv[a]
160	59.7 ± 1.7	51.0 ± 1.8	40.2 ± 1.3	70.5 ± 1.6	48.8 ± 1.3	59.9 ± 1.1	50.3 ± 1.5	60.8 ± 1.5
240	54.2 ± 1.1	41.8 ± 1.4	10.8 ± 0.8	68.7 ± 1.0	46.2 ± 1.3	57.6 ± 0.9	47.0 ± 1.3	60.1 ± 1.3
320	52.4 ± 1.0	29.1 ± 1.1	20.9 ± 0.9	66.4 ± 1.1	43.7 ± 0.8	57.3 ± 0.9	44.8 ± 0.8	59.7 ± 0.8

[a]LDA is used for dimensionality reduction prior to classification

Table 4 The p-values and h-values of Wilcoxon test based on ORL face database.

Method	10 times		20 times		30 times	
	p-value	h-value	p-value	h-value	p-value	h-value
ELM vs. E-ELMcv	<0.001	1	<0.001	1	<0.001	1
ELM vs. IE-ELMcv	<0.001	1	<0.001	1	<0.001	1

of learning/classification rather than dimension reduction while most face recognition techniques are approaches for feature extraction by reducing feature dimension, thus a direct comparison between classical face recognition methods and ELM based methods may not provide meaningful information. We therefore further investigated the use of dimension reduction + proposed methods (E-ELMcv[a] and IE-ELMcv[a]) and found that these new learning techniques achieved higher classification accuracy than Bayes, LDA and ULDA methods. Better classification performance can be expected by replacing LDA with more sophisticated dimension reduction methods prior to applying ELM based classifiers.

4.3 Statistical Analysis on Stability

We have conducted statistical analysis following the work suggested in Zhai et al. [26] to analyze stabilities of our proposed methods. Wilcoxon test and paired t-test [4] were used and the ORL face database was adopted for the analysis. By running ELM, E-ELMcv and IE-ELMcv for 10, 20 and 30 times, we obtained nine statistics denoted as M_i^1, M_i^2 and M_i^3 ($i = 1, 2, 3$), which are corresponding to ELM, E-ELMcv and IE-ELMcv. Parameter i represents the number of runs, e.g. M_1^1 is the statistics obtained by running ELM for 10 times. We aim to compare the performance between ELM and our proposed methods, therefore we compute two sets of results for ELM vs. E-ELMcv and ELM vs. IE-ELMcv as shown in Tables 4 and 5. As suggested in Zhai et al. [26], we used MATLAB functions `ranksum` and `ttest2` for calculating Wilcoxon test and t-test statistics, respectively. The small p-values (< 0.001) for both tests further demonstrated the effectiveness of our proposed methods. Furthermore, we analyzed the stability of our methods with coefficient of variation (CV) of testing accuracy. The coefficient of variation is calculated as follows

$$CV = \sigma/\mu \tag{13}$$

where σ is the standard deviation of the testing accuracy across 20 runs and μ is the mean testing accuracy. We evaluate our methods with four different hidden node number and the results are shown in Table 6. It is observed from the comparison results that the proposed E-ELMcv and IE-ELMcv are more stable than ELM in terms of achieving smaller CV values.

4.4 The Effects of Parameter Selection

The effects of parameters on generalization performances in E-ELMcv are depicted in Fig. 3. It can be observed from Fig. 3a that if five or more folds are applied, the classification accuracies will be larger than that of ELM and increase steadily. A small fold number results in poor generalization performance because less training samples involve in the learning process. Figure 3b shows that a large number of hidden nodes might give higher accuracies in testing. However, a complex network could also overfit the training data. For example, when the number of hidden nodes is larger than 80, the generalization performance decreases a lot.

In the IE-ELMcv algorithm, e_2 serves as a major factor to control the ranges of N_u. In the experiments, e_2 is set to 5 by default. When the value of e_2 is reduced to 2, the corresponding result on ORL database is 84.95 ± 1.49 in percentage. Obviously, a small e_2 (more "invariant" hidden nodes) can lead to more satisfactory performances in classification, possibly because the network complexity is simplified. In other words, redundancy among the

Table 5 The p-values of t-test based on ORL face database.

Method	10 times	20 times	30 times
ELM vs. E-ELMcv	<0.001	<0.001	<0.001
ELM vs. IE-ELMcv	<0.001	<0.001	<0.001

Table 6 Comparison of stability between the proposed methods and ELM.

Nodes	ELM	E-ELMcv	IE-ELMcv
25	0.0626	0.0401	0.0616
50	0.0362	0.0290	0.0312
75	0.0294	0.0214	0.0209
100	0.0347	0.0213	0.0258

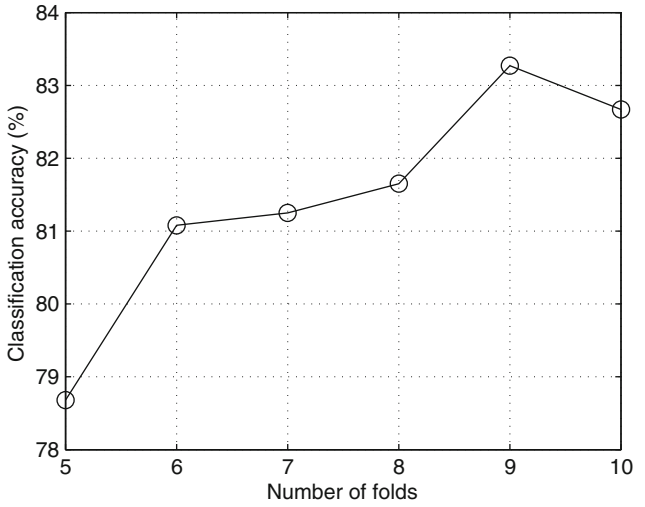

Figure 3 Results on ORL database using E-ELMcv method: **a** Classification results with different number of folds for cross-validation where $\tilde{N}$ is 100; **b** classification results with different number of hidden nodes where R is 10.

input weights and hidden node biases are removed by assigning constant values to the parameters of the selected hidden nodes.

5 Conclusion

In this paper, the cross-validation strategy is introduced into the training process of E-ELM algorithm to avoid the overfitting problem and increase the generalization performance. As a result, E-ELMcv and IE-ELMcv are proposed and validated for image analysis. The experimental results demonstrate that our proposals outperform the conventional E-ELM algorithm in terms of classification accuracy. Although the proposed methods need more training time than ELM does, they are still effective when compared with E-ELM and traditional gradient-based learning algorithm. In addition, it is also possible to alleviate the computational burden by selecting proper network parameters.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, J.D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 711–720.
2. Chen, L., Man, H., Nefian, V.A. (2005). Face recognition based on multi-class mapping of fisher scores. *Pattern Recognition*, *38*, 799–811.
3. Chen, W.L., Er, M.J., Wu, S.Q. (2006). Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *36*, 458–466.
4. Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.
5. Duda, R.O., Hart, P.E., Stork, D.G. (2001). *Pattern classification*. New York: Wiley.
6. Graham, D.B., & Allinson, N.M. (1998). Characterizing virtual eigensignatures for general purpose face recognition In H. Wechsler, P.J. Phillips, V. Bruce, F. Fogelman-Soulie, T.S. Huang (Eds.), *Face recognition: From theory to applications, NATO ASI series F, computer and systems sciences* (Vol. 163, pp. 446–456).
7. Hafed, Z.M., & Levine, M.D. (2001). Face recognition using the discrete cosine transform. *International Journal of Computer Vision*, *43*, 167–188.
8. Huang, G.B., Chen, L., Siew, C.K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, *17*, 879–892.
9. Huang, G.B., Li, M.B., Chen, L., Siew, C.K. (2008). Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing*, *71*, 576–583.
10. Huang, G.B., Wang, D.H., Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*, 107–122.
11. Huang, G.B., Zhu, Q.Y., Siew, C.K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*, 489–501.
12. Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 491–502.
13. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N. (2009). Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition. *IEEE Transactions on Neural Networks*, *20*, 103–123.
14. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N. (2005). Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, *26*, 181–191.
15. Moghaddam, B., Jebara, T., Pentland, A. (2000). Bayesian face recognition. *Pattern Recognition*, *33*, 1771–1782.
16. Nixon, M., & Aguado, A.S. (2008). *Feature extraction and image processing*. Oxford: Academic Press.
17. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J. (2000). The FERET evaluation methodology for face-recognition algorithms.

IEEE Transactions on Pattern Analysis and Machine Intelligence, *22*, 1090–1104.
18. Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*, 21–45.
19. Rong, H.J., Ong, Y.S., Tan, A.H., Zhu, Z.X. (2008). A fast pruned-extreme learning machine for classification problem. *Neurocomputing*, *72*, 359–366.
20. Samaria, F.S. (1994). Face recognition using hidden markov models. PhD thesis, University of Cambridge, Cambridge.
21. Serre, D. (2002). *Matrices: Theory and applications*. New York: Springer.
22. Storn, R., & Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*, 341–359.
23. Wang, X.Z., & Dong, C.R. (2009). Improving generalization of fuzzy if–then rules by maximizing fuzzy entropy. *IEEE Transactions on Fuzzy Systems*, *17*, 556–567.
24. Wang, X.Z., Zhai, J.H., Lu, S.X. (2008). Induction of multiple fuzzy decision trees based on rough set technique. *Information Sciences*, *178*, 3188–3202.
25. Ye, J. (2005). Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, *6*, 483–502.
26. Zhai, J.H., Xu, H.Y., Wang, X.Z. (2012). Dynamic ensemble extreme learning machine based on sample entropy. *Soft Computing*, *16*, 1493–1502.
27. Zhang, R.X., Huang, G.B., Sundararajan, N., Saratchandran, P. (2007). Multicategory classification using extreme learning machine for microarray gene expression cancer diagnosis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*, 485–495.
28. Zhu, Q.Y., Qin, A.K., Suganthan, P.N., Huang, G.B. (2005). Evolutionary extreme learning machine. *Pattern Recognition*, *38*, 1759–763.

Nan Liu received the BEng degree in electrical engineering from University of Science and Technology Beijing, China, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore. He is currently a Senior Research Scientist at the Department of Emergency Medicine, Singapore General Hospital. His research interests include pattern recognition, machine learning, and biomedical signal processing.

Han Wang received the BEng degree from Northeastern Heavy Machinery Institute, China, and the PhD degree from Leeds University, UK. He is currently an Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. After receiving the PhD degree, he joined the Robotics Research Group, Oxford University, UK, as Research Office for three years. He was also with Carnegie Mellon University, USA, and Monash University, Australia, as a Visiting Scientist. His research interests include computer vision, evolutionary computing and robotics.