Cogn Comput
DOI 10.1007/s12559-013-9224-1

Fast Face Recognition Via Sparse Coding and Extreme Learning Machine

Bo He · Dongxun Xu · Rui Nian · Mark van Heeswijk · Qi Yu · Yoan Miche · Amaury Lendasse

Received: 9 August 2012 / Accepted: 4 March 2013
© Springer Science+Business Media New York 2013

Abstract Most face recognition approaches developed so far regard the sparse coding as one of the essential means, while the sparse coding models have been hampered by the extremely expensive computational cost in the implementation. In this paper, a novel scheme for the fast face recognition is presented via extreme learning machine (ELM) and sparse coding. The common feature hypothesis is first introduced to extract the basis function from the local universal images, and then the single hidden layer feedforward network (SLFN) is established to simulate the sparse coding process for the face images by ELM algorithm. Some developments have been done to maintain the efficient inherent information embedding in the ELM learning. The resulting local sparse coding coefficient will then be grouped into the global representation and further fed into the ELM ensemble which is composed of a number of SLFNs for face recognition. The simulation results have shown the good performance in the proposed approach that could be comparable to the state-of-the-art techniques at a much higher speed.

Keywords Extreme learning machine · Common feature hypothesis · Sparse coding · Face recognition

B. He · D. Xu · R. Nian (✉)
School of Information Science and Engineering,
Ocean University of China, Qingdao 266100,
Shandong, China
e-mail: nianrui_80@163.com

B. He
e-mail: bhe@ouc.edu.cn

M. van Heeswijk · Q. Yu · Y. Miche · A. Lendasse
Department of Information and Computer Science,
Aalto University, 00076 Aalto, Finland
e-mail: mark.van.heeswijk@aalto.fi

Q. Yu
e-mail: qi.yu@aalto.fi

Y. Miche
e-mail: yoan.miche@aalto.fi

A. Lendasse
IKERBASQUE, Basque Foundation for Science,
48011 Bilbao, Spain
e-mail: amaury.lendasse@aalto.fi

A. Lendasse
Computational Intelligence Group, Computer Science Faculty,
University of the Basque Country, Paseo Manuel Lardizabal 1,
Donostia/San Sebastián, Spain

Introduction

Cognitive science is the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, brain theory, linguistics, and anthropology [1]. Over the past few decades, there is a variety of the cognitive-inspired computation for the image processing and understanding [2–4]. Face recognition remains among the most challenging research topic in the cognition computation field. The main reasons include the highly overlapping intra- and inter-identity distributions due to the pose, age, expression, occlusion, and the external imaging factors such as the variations of illumination.

In general, there are two steps in a face recognition system. The first step is to define an effective representation of the face images, which contains sufficient information for the future classification. The second step is to classify a new face image with the chosen representation. The approaches to face recognition can be basically divided into three categories [5]: global or holistic approach,

Published online: 13 July 2013

local approach, and hybrid approach. The global approach utilizes the entire face image information to construct features for recognition [6]. This approach shows relatively good performance for face images of the frontal view [6, 7]. However, it can be sensitive to variations that resulted from the imaging factors. On the other hand, the local approach does not suffer much from the imaging factors since such variations affect the face image only partially [8]. Some local methods also adopt blocks of appearances such as the regions of the eyes, the mouth, and the nose as the local features [9, 10]. Although the local approach works well over the global approach for those face images with variations [10, 11], most of the local methods are required to locate the positions of the facial components and extract them for feature construction [10–12]. The hybrid approach utilizes both global and local facial information and can be considered to be similar to that of human's recognition process, but it may bring a high computational cost.

Most of the approaches described above for face recognition focus on the sparse coding. In the human vision system, when the light falls on the retina at the back of the eye, it converts into the electrical pulses immediately. After various neural layers in the retina, the signals pass to the lateral geniculate nucleus (LGN). The striate cortex structure is at the back of the brain, which is the primary visual cortex, V1. The neurons in V1 take each input from a number of geniculate neurons, and any individual neuron can only see a small portion of the image that the eyes are viewing. This small region is the receptive field and can be characterized as being localized, oriented, and bandpass [13]. Olshausen and Field [14, 15] have indicated that the neural networks in the human vision system could perform *sparse coding* of the learned features qualitatively similar to the receptive fields of simple cells in V1. Sparse coding provides a class of algorithms for finding the succinct representations of the stimuli. Given the unlabeled input data only, it learns the basis functions that capture the higher-level features. When a sparse coding algorithm is applied to natural images, the learned bases resemble the receptive fields of the neurons in the visual cortex [14–16]. Moreover, sparse coding produces the localized bases when applied to other natural stimuli such as the speech and the video [17, 18]. Sparse coding can be applied to learn overcomplete basis sets, and model inhibition between the bases by sparsifying the activations. Efficient sparse coding algorithms were also discussed by iteratively solving the L_1-regularized least squares problem and the L_2-constrained least squares problem [16]. For the face recognition, Wright et al. [19] proposed to apply sparse coding and achieved an impressive recognition performance. Huang et al. [20] proposed a new sparse coding recovery method that is invariant to image-plane transformation to deal with the misalignment and pose variation in face recognition. Wagner et al. [21] presented a sparse representation-based method that could deal with face misalignment and illumination variation, and Yang and Zhang [22] used Gabor features to reduce greatly the size of occlusion dictionary and got a higher accuracy. However, most of the approaches to face recognition use the face data sets to learn the basis function. Shan [23] developed a hierarchical model, recursive ICA (RICA), which captures nonlinear statistical structures of the visual inputs that cannot be captured by a single layer of ICA. Inspired by that, Shan [24] then carried out different recognition tasks by sparse coding learned from the natural images. However, the development of sparse coding models has been hampered by their expensive computational cost. In particular, learning large, highly overcomplete representations has been extremely expensive.

Recently, extreme learning machine (ELM) has attracted more and more attention in machine learning by providing the higher generalization performance at a much faster speed [25, 26]. ELM was originally developed for the single hidden layer feedforward networks (SLFN) instead of the classical gradient-based algorithms [25–27], then extended to the generalized SLFN that need not be the neuron alike [28, 29], and can work for the conventional SVM and its variants. The essence of ELM is that: When the input weights and the hidden layer biases are randomly assigned, the output weights can be computed by the generalized inverse of the hidden layer output matrix [25, 26]. There are a great many ELM variations that have been proposed, including the random hidden layer feature mapping-based ELM [30], the Kernel-based ELM [30–32], the fully complex ELM [33], the incremental ELM (I-ELM) [27–29], the online sequential ELM (OS-ELM) [34–36], the pruning ELM (P-ELM, OP-ELM) [37, 38], the circular-ELM (C-ELM) [39], ELM ensembles [40, 41], etc., which have led to the state-of-the-art results in many applications both for the regression and for the pattern recognition problem [42–46].

In this paper, we come up with a new fast face recognition algorithm via sparse coding and ELM. We firstly set up the common feature hypothesis to focus on the sparse coding of the local universal images and then extract the basis function in common. The image patches and the corresponding sparse coding coefficients in the last iterative step are then taken into the established SLFN so that the whole learning process could be simulated by the ELM algorithm for the face images. With some developments in the high-dimensional space such as the whitening, the principal component analysis (PCA), and the nonlinear transformations, the resulting local sparse coding coefficients will be organized into a global representation and further fed into the ELM ensemble for face recognition.

The rest of the paper is organized as follows. In "The Basics of ELM" section and "Sparse Coding" section, we will first introduce the basic theory of ELM and sparse coding. The "Fast Face Recognition" section will describe the whole process of our proposed method in detail. Simulation and result analysis will be shown in the "Simulations and Result Analysis" section. Finally, "Conclusion" section will make a conclusion for the paper.

The Basics of ELM

So far, ELM learning has been developed to work at a much faster learning speed with the higher generalization performance, both in the regression problem and in the pattern recognition. For the given N learning samples $\{x_i, y_i\}_{i=1}^{N}$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^{'}$ and $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^{'}$, the standard model of the ELM learning can be written as the following matrix format:

$$
\begin{aligned}
&\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \\
&\mathbf{H}(x) = [h_1 \quad h_2 \quad \cdots \quad h_L] \\
&= \begin{bmatrix} h_1(x_1) & . & . & . & h_L(x_1) \\ . & & . & & . \\ . & & . & & . \\ . & & . & & . \\ h_1(x_N) & . & . & . & h_L(x_N) \end{bmatrix} \\
&= \begin{bmatrix} g(\omega_1 \times x_1 + b_1) & . & . & . & g(\omega_L \times x_1 + b_L) \\ . & & . & & . \\ . & & . & & . \\ . & & . & & . \\ g(\omega_1 \times x_N + b_1) & . & . & . & g(\omega_L \times x_N + b_L) \end{bmatrix}_{N \times L} \\
&\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_L]^{'}_{m \times L}, \quad \mathbf{Y} = [y_1, y_2, \ldots, y_N]^{'}_{m \times N}
\end{aligned}
\tag{1}
$$

where $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}]^{'}$ is the weight vector connecting the ith hidden neuron and the input neurons, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots \beta_{im}]^{'}$ denotes the weight vector connecting the ith hidden neuron and the output neurons, and there are L hidden neurons with the activation function $g(x)$. All kinds of the activation functions can be chosen here, such as the Sigmoid function, the hard-limit function, the Gaussian function, the multiquadric function, and so on.

If the activation function $g(x)$, ω, and b are all set, the only learning parameter will be $\boldsymbol{\beta}$. Different from the traditional learning algorithm, ELM tends to achieve the least training error and the least norm of output weight together. According to Bartlett's theory [47], when the feedforward neural networks get smaller training error, the norms of weights are smaller, and the generalization performance of the networks is better, $\boldsymbol{\beta} = \arg\min(||\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}||^2, ||\boldsymbol{\beta}||)$. In order to solve the formation, both the standard optimization method and the minimal norm least square method need to be adopted. The original implementation of ELM will then be $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{Y}$, where $\mathbf{H}^{\dagger}$ denotes the Moore–Penrose generalized inverse of matrix H [25]. The orthogonal projection method can be used here when $\mathbf{H}^T\mathbf{H}$ is nonsingular and $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, or when $\mathbf{H}\mathbf{H}^T$ is nonsingular and $\mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}^T\mathbf{H})^{-1}$. In addition, the resulting solution tends to be more stable with better generalization performance by adding a positive value to the diagonal of $\mathbf{H}\mathbf{H}^T$ or $\mathbf{H}^T\mathbf{H}$.

Sparse Coding

Sparse coding was first coined by Olshausen and Field [14, 15], which attempts to find the sparse linear representations of the certain image with respect to an overcomplete dictionary and provide more efficient sparse coding, similar to the primary visual cortex in the human visual system. The basic model of the sparse coding can be denoted as a sparse linear superposition of the basis functions,

$$X = DA \tag{2}$$

where $X = [x_1, x_2, \ldots, x_d]^{'} \in R^d$ represents the image patch in a d-dimensional space, $A = [a_1, a_2, \ldots, a_t]^{'} \in R^t$ corresponds to the sparse representation coefficients of the original image patch X in a t-dimensional space, and D refers to a $d \times t$ matrix of the basis functions. The goal of the sparse coding is to find the basis function matrix D so that the dynamic coefficient values A can be as statistically independent as possible over an ensemble of the images, and bear the sparse structure, i.e., a specific low entropy code where the probability distribution of each coefficient's activity is unimodal and peaked around zero.

Generally, sparse coding can be formulated as the optimization problem by minimizing the following cost function E, $E = \|X - DA\|_2^2 + \lambda\|A\|_1$, where the sparse coding will be obtained by $\min||A||_1$, s.t. $\|X - DA\|_2^2 \leq \varepsilon$, the L_1 norm $||A||_1$ is to enforce sparseness, the L_2 norm constraint $\|X - DA\|_2$ on the columns of D can remove the scaling ambiguity, and λ is a positive Lagrange multiplier that determines the importance of the second term relative to the first one. By imposing L_1 norm regularization on representation coefficients, sparse coding can be solved efficiently [15]. As sparse coding needs to encode a large set of image patches, the bottleneck is mainly the computational speed.

Fast Face Recognition

General Learning Model

The idea of the fast face recognition here is that we try to apply one common feature hypothesis into the basis function of the sparse coding in common from the universal image patches instead of directly from face image patches by means of ELM learning in SLFN. The flowchart of our approach is shown in Fig. 1, where the left part represents the learning process of the important parameters extracted from the universal images, and the right part is referred as the test process for face recognition. The key points in our approach are the sparse coding from the universal image patches and the ELM learning of the face image patches.

Before formally simulating the basis function D in the sparse coding of the universal images, some preprocessing has been first done, such as dividing the images into patches, whitening, and the dimensionality reduction. A number of common visual features can then be extracted from randomly collected universal images after being adjusted by one nonlinear transformation, and the universal image patches $X_0 = X$ and the underlying sparse representation $A_0 = A$ will be acquired in the iteration process at the same time. The embedding basis function D of the universal images will be simulated by the ELM algorithm

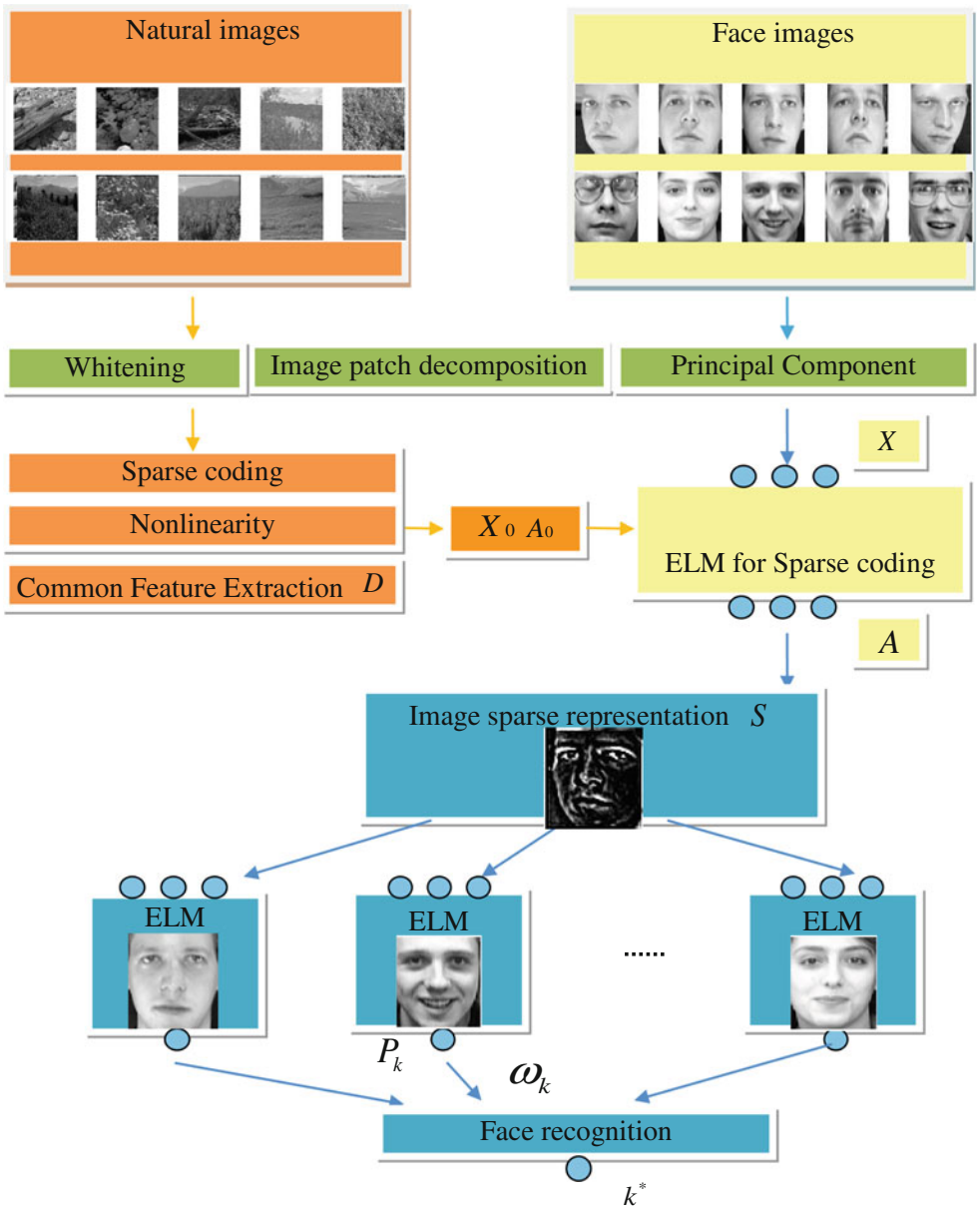

Fig. 1 *Flowchart* of our approach

in the learning process so that we can make efforts to achieve the sparse coding vectors A from the image patches X for the face images once the SLFN for simulation is reasonably established. The resulting sparse coding representation S of each face image will be further fed into the next SLFN in the test process for face recognition. The decision k^* will be made in an ensemble, by the output P_k and the importance weight $\omega_k$for the individual SLFN of the kth person.

Common Feature Hypothesis

Let there be K persons $\{P_1, \ldots, P_k, \ldots, P_K\}$; each person P_k corresponds to Q face images $\{I_1^k, \ldots, I_q^k, \ldots, I_Q^k\}$, where I_q^k denotes the qth image of the given person P_k. For the human visual system, one notable advantage is that human beings can recognize one person at a simple glance of only a few or even one of the face images, while most face recognition approaches developed so far in the computer vision system depend on a huge number of face images for learning, which are poorly available when the image collection is generally expensive.

With a deep analysis of the human visual system, it is found that the low-level visual layers, such as retina, LGN and V1 (the primary visual cortex), are shared components that process all the visual information we perceive. These layers develop and mature gradually since childhood and provide the basis with common features from the scenes encountered for all the visual tasks in life.

Therefore, the concept of the common feature hypothesis suggests that all visual stimuli share characteristics in common such that the knowledge from one set of visual stimuli can be applied to a completely different one. So, here, we try to extract those common visual features that are essential for face recognition from a set of the universal images, e.g., the nature images, and provide the information for the ELM learning in the next step. Suppose that the number of the natural images is n, $\{I_1, \ldots, I_i, \ldots, I_n\}$, there must be some inherent common visual features D that can be extracted both in the natural images and in the face images.

$$\begin{aligned} &F_{\text{nature}} \subseteq f(I_1, \ldots, I_i, \ldots, I_n), \\ &F_{\text{face}} \subseteq f\left(I_1^k, \ldots, I_q^k, \ldots, I_Q^k\right), \quad D \in \{F_{\text{nature}} \cap F_{\text{face}}\} \end{aligned} \tag{3}$$

where f denotes the attribution extraction function, F_{nature} and F_{face} are, respectively, the typical features obtained from the output of the function f by the natural images and face images, and D represents those knowledge that are shared by the different sources of the visual stimuli. One example of the common feature hypothesis is shown in Fig. 2, where the left column are, respectively, one natural image and one face image, and Fig. 2b, d are the corresponding top eigenvectors when sampling image patches and applying PCA to the images. Although the natural image and the face image display different visual contents, they share very similar local statistical structure here.

Sparse Coding with ELM

Whitening

Suppose the size of each natural image is $M \times M$, the natural images are first transformed by a whitening filter and then normalized to follow a Gaussian function with the zero-mean vector and the unit variance. It is believed that a surprising fact in the human visual system is that there does exist the marginal distribution regularization process and the sensory inputs are whitened in the retina and the LGN before the transmission to V1. Besides the functional role of removing the second-order pairwise redundancy as the natural images obey the $1/f$ power law in the frequency domain, whitening might also serve as formatting the sensory input for the cortex so that the basis function could cover a broad range of spatial frequencies. The steps of the whitening process are as follows. To avoid the boundary effects, before dividing the natural images into all the possible image patches, we will first cut a number of pixels m off the boundary to change each image into a $(M - 2\,m) \times (M - 2\,m)$ size. Afterward, we make a subtraction

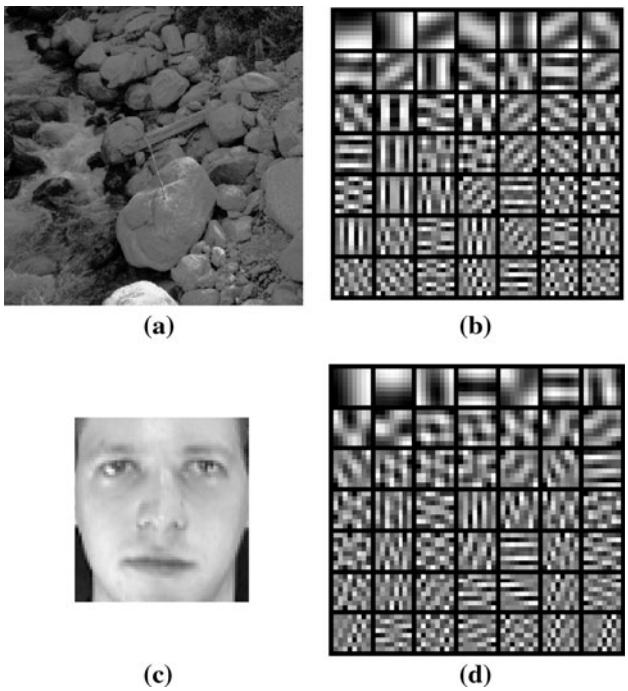

Fig. 2 Common feature hypothesis. **a** Natural image. **b** Top eigenvectors. **c** Face image. **d** Top eigenvectors

of each image patch in a range of $p \times p$ size by the local mean, and the PCA projection matrix PJ will be further calculated. In this way, each image patch can be represented as a $(p^2 - 1)$-dimensional vector x and then be scaled to the unit variance. Figure 3 shows one example of the whitening process, which regulates the marginal distribution of the original image to follow a generalized-Gaussian-like distribution.

Sparse Coding

The basis function D will be initialized with the Gaussian random variables, and then, on each iteration, we randomly pick N image patches to form $X = (x_1, x_2, \ldots, _N)$ after PCA projecting. Assuming the sparse coding coefficients, $A = (a_1, a_2, \ldots, a_N)$ follow a marginal prior as follows:

$$P(a_j) \propto \exp\left(-\gamma \Phi(a_j)\right) \tag{4}$$

where a_j refers to the jth dimension for each coefficient, γ is a scaling constant, and Φ refers to the sparsity function that can be taken as $\Phi(a_j) = \left\|a_j\right\|_1$ by imposing L_1 norm. The above nonlinear function will assess the sparseness of the code for the given natural image by assigning a cost depending on how activity is distributed among the coefficients. The cost function we construct for the sparse coding takes the sum of each coefficient's activity to meet the criterion and the choice of the nonlinear function will favor among activity states with the fewest number of nonzero coefficients. The new basis function D will then come into being from the corresponding sparse representation A for the next iteration,

$$D = D + \frac{\eta}{N} \sum_{i}^{N} (x_i - Da_i) a_i^{'} \tag{5}$$

where η denotes the learning rate, and a_i is the most probable underlying signal given the image patch observation x_i and the current basis function D. After each update, the columns of A are normalized to unit length to speedup the learning process.

When the number of the iteration steps approaches what we allocate in advance, the final basis function D, the image patches X_0, and the corresponding sparse representation A_0 will be obtained from the natural images in the end, where $X_0 = X$ is a collection of the N natural image patches and $A_0 = A$ is the underlying sparse representation of all the extracted image patches in the last iteration step.

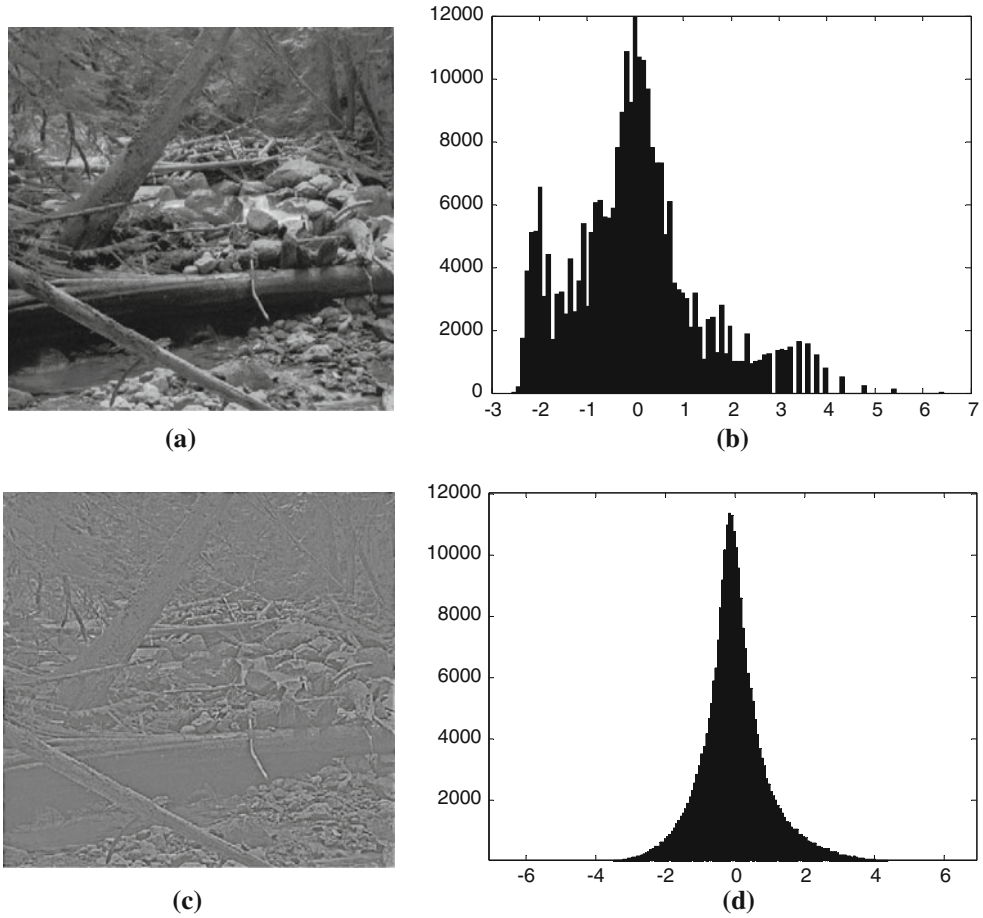

Fig. 3 Distribution of the pixel values in the whitening process. **a** Original image. **b** Distribution of the original image. **c** Image by the whitening transform. **d** Distribution in the whitened image

In this way, X_0 takes on a $N \times (p^2 - 1)$ matrix and A_0 a $N \times p^2$ matrix, respectively.

Nonlinearity

A further development to convert the highly sparse distribution of the output A into a Gaussian distribution will be introduced here with the utility of the component-wise nonlinear function. For each dimension a_j of A, the empirical cumulative distribution function (CDF) of the absolute value $|a_j|$ will be first estimated by calculating the histogram of $|a_j|$ in the range of the bins between $b_{\min}$ and $b_{\max}$, and the size of each $b_{\min}$ is b. After this, the CDF function will then be fitted as follows:

$$F_j(|a_j|) = \Gamma\left((|a_j|/\tau)^{\theta}, 1/\theta\right) \tag{6}$$

where Γ denotes the incomplete Gamma function, $\Gamma(x, y) = \int_y^{+\infty} t^{x-1}\mathrm{e}^{-t}\mathrm{d}t$, $\theta > 0$ is a shape parameter, and $\tau > 0$ is a scale parameter. The function F can further modify the sparse coding coefficients to help distinguishing by the coordinate-wise activation function $G(a_j) = g(F(|a_j|))$, where g denotes the inverse CDF function of a standard normal distribution. In this way, the activation function discards the signs of the outputs and converts the marginal distributions to the Gaussian distributions, so that the common feature extracted from the natural images can further be embedded into the ELM learning.

ELM Learning

During the above learning process, in addition to that, the basis function D will be derived from the natural images for the common feature extraction. Simultaneously, after taking a series of the iterative steps, the latest image patches X_0 and the corresponding sparse coding representations A_0 will also be obtained, which can be denoted as the image package $P_0 = \{X_0, A_0\}$ for an easy expression. Similarly, all the face images will also be firstly preprocessed and whitened as the natural images, and after PCA projecting, all the N' possible image patches will be collected to constitute $X = (x_1, x_2, \ldots, x_{N'})$ from the face images. Suppose that there are N arbitrary distinct training samples $P_0 = \{x_{0,i}, a_{0,i}\}_{i=1}^{N}$, with the input $x_{0,i} = [y_{i,1}, y_{i,2}, \ldots, y_{i,p\times p-1}]' \in R^{p\times p-1}$ and the expected output $a_{0,i} = [z_{i,1}, z_{i,2}, \ldots, z_{i,p\times p}]' \in R^{p\times p}$. We take the natural image patches X_0 as the inputs for SLFN to predict the sparse coding coefficients of the face image patches with the underlying common feature strategy conducted by ELM:

$$\begin{aligned}
&\mathbf{H}\boldsymbol{\beta} = A_0\\
&\mathbf{H}(\omega_1, \ldots, \omega_L, b_1, \ldots, b_L, x_{0,1}, \ldots, x_{0,N})\\
&= \begin{bmatrix} g(\omega_1 \times x_{0,1} + b_1) & \cdots & g(\omega_L \times x_{0,1} + b_L) \\ \vdots & g(\omega_i \times x_{0,j} + b_i) & \vdots \\ g(\omega_1 \times x_{0,N} + b_1) & \cdots & g(\omega_L \times x_{0,N} + b_L) \end{bmatrix}_{N\times L}\\
&\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_L]'_{p^2\times L}, \quad A_0 = [a_{0,1}, \ldots, a_{0,N}]'_{p^2\times N}
\end{aligned} \tag{7}$$

where every input $x_{0,j}$ is composed of the natural image patch and each expected output is the sparse coding representation $a_{0,j}$. The SLFN is established and initialized to learn the sparse coding process from the natural images. When we feed the SLFN with the face image patches X, the actual output will be considered as the estimation of the corresponding sparse coding coefficients:

$$a_j = \sum_{i=1}^{L} \beta_i g(\omega_i \times x_j + b_i), \quad j = 1, \ldots, N' \tag{8}$$

For the face recognition problem, theoretically, the model selection of the ELM architecture could be evaluated by the generalization error as follows:

$$E = \lim_{N'\to\infty} \sum_{j=1}^{N'} \left(F(\boldsymbol{x}_j) - \boldsymbol{a}_j\right)^2 / N', \quad j = 1, \ldots, N' \tag{9}$$

where F is the input–output function of the ELM learning, $\boldsymbol{x}_j$ is the face image patch, $F(\boldsymbol{x}_j) = \hat{\boldsymbol{a}}_j$ is the real output of the SLFN corresponding to input, and a_j is the expected output. In practice, the leave-one-out (LOO) cross-validation could be a good choice for the model selection in SLFN, which is basically a special case of k fold cross-validation in the case where $k = N'$. All the face image patches are divided into N' parts for the training sets; in each one, there is exactly one sample that has been left out for testing, and then the estimation of the generalization error becomes

$$E = \sum_{j=1}^{N'} \left(F(\boldsymbol{x}_j, -j) - \boldsymbol{a}_j\right)^2 / N', \quad j = 1, \ldots, N' \tag{10}$$

where $F(\boldsymbol{x}_j, -j)$ denotes the output of the jth training sets without the jth sample. The parameters Θ such as the size of the image patch $p \times p$ and the number of the hidden neurons L will be chosen by the minimum generalization error $\Theta = \arg\min_{\Theta} E$.

Face Recognition with ELM

For the face recognition, suppose that the number of all the face images is $Q' = K \times Q$, the size of each face image is $M' \times M'$, the training set is composed of the Q'_1 labeled images from the known persons, and the rest Q'_2 face

images unlabeled constitute the test set, $Q_1^{'} + Q_2^{'} = Q^{'}$. The total amount of the sparse coding representations $A = (a_1, a_2, \ldots, a_{N'})$ is $N^{'} = Q^{'} \times M^{'} \times M^{'}/p \times p$. During the test process, on the basis of the prior knowledge getting from the ELM learning by the decomposition of the face images, we further feed the next SLFN to perform face recognition by ELM algorithm:

$$\begin{aligned} &\mathbf{H}\boldsymbol{\beta} = P \\ &\mathbf{H}\left(\omega_1, \ldots, \omega_L, b_1, \ldots, b_L, S_1, \ldots, S_{Q_1^{'}}\right) \\ &= \begin{bmatrix} g(\omega_1 \times S_1 + b_1) & \cdots & g(\omega_L \times S_1 + b_L) \\ \vdots & g(\omega_i \times S_j + b_i) & \vdots \\ g(\omega_1 \times S_{Q_1^{'}} + b_1) & \cdots & g(\omega_L \times S_{Q_1^{'}} + b_L) \end{bmatrix}_{Q_1^{'} \times L} \\ &\boldsymbol{\beta} = \left[\beta_1, \beta_2, \ldots, \beta_L\right]_{K \times L}^{'}, \quad P = \left[P_1, \ldots, P_k, \ldots, P_K\right]_{K \times Q_1^{'}}^{'} \end{aligned} \tag{11}$$

where S_j denotes the sparse coding representation which is made up of the coefficients $\{a_1, a_2, \ldots, a_{N'/Q'}\}$ for the image patches derived from the given jth face image I_j in the training set. The recognition process is to specify the membership of the face image to be recognized by the mapping with respect to the ELM learning:

$$P_j = \sum_{i=1}^{L} \beta_i g(\omega_i \times S_j + b_i), \quad j = 1, \ldots, Q_2^{'} \tag{12}$$

where S_j corresponds to the sparse coding representation of the face image I_j in the test set, and $P_j = \left[P_{j1}, \ldots, P_{jk}, \ldots, P_{jK}\right]^{'}$ is the output of the SLFN from the input S_j.

ELM ensemble can be further set up here to improve generalization performance, with every person an individual SLFN. Hansen and Salamon [48] showed that the performance of a single neural network can be expected to improve by the ensemble with a plurality consensus scheme. The underlying is to generate multiple versions of recognition process for the same task which when combined will improve the interpretation and provide more stable predictions about the faces. The learning process in the ELM ensemble is to take a linear fusion strategy of the individual outputs. The decision will be made by $k^* = \arg\max_{k=1}^{K} \omega_k P_k$, where K is number of the persons, P_k is the output of kth SLFN, and ω_k corresponds to the importance weight, $0 \leq \omega_k \leq 1$. In this case, the output of each individual can be pruned to $P_k \in R$ for simplification, and the combination weights could all set to be equal when face images from every person make the same contribution to the optimal cognition course.

Fast Face Recognition Algorithms

The efficient sparse coding is introduced here to apply on the natural image patch vectors [16]. When the standard generative model assumes that $X - DA$ is distributed as a zero mean Gaussian distribution, the algorithm is as follows:

Algorithm 1 Basic sparse coding

1. Input all the image patches and set the parameter λ.
2. Initialize the basis function matrix D, the sparse coding matrix A with a Gaussian random matrix
3. For $t = 1$: T
   - Random select N image patches $X = (x_1, x_2, \ldots, x_N)$ from all image patches
   - For $t = 1$: N
     - Apply the feature-sign search algorithm
     - Compute the sparse coding coefficient
   - $a_i = \arg\min \|x_i - Da_i\|_2^2 + \lambda \|a_i\|_1$ for each image patch x_i
   - End
   - Update the basis function matrix $D^T = (a_i a_i^T + \Lambda)^{-1} (x_i a_i^T)^T$ where
   - $\Lambda = \mathrm{diag}(\lambda)$
   - Get the sparse coding matrix $A = (a_1, a_2, \ldots, a_N)$
   - End
4. Set $X_0 = X$, $A_0 = A$, D as the outputs

The feature-sign search algorithm maintains an active set of the potentially nonzero coefficients and their corresponding signs and systematically searches and converges to the optimal solution [16].

Algorithm 2 Feature-sign search

1. Initialize the sparse coding coefficient $a = \vec{0}$, the feature-sign vector $v = \vec{0}$, and the active set $U = \{\}$, where $v_i \in \{-1, 0, 1\}$ denotes the sign of a_j
2. From the zero coefficients of a, select $j = \arg\max_j \left|\frac{\partial \|x - Da\|^2}{\partial a_j}\right|$ and activate a_j only if it locally improves the objective
   - If $\frac{\partial \|x - Da\|^2}{\partial a_j} > \lambda$, set $v_j = -1, \quad U = \{j\} \cup U$
   - If $\frac{\partial \|x - Da\|^2}{\partial a_j} < -\lambda$, set $v_j = -1, \quad U = \{j\} \cup U$
3. Set $\widehat{D}$ as a submatrix of D that contains only the columns corresponding to U
   - Set $\widehat{a}$ and $\widehat{v}$ as the subvectors of a and v corresponding to U
   - Compute the analytical solution to the resulting unconstrained quadratic optimization problem (QP), $\widehat{a}_{\mathrm{new}} = (\widehat{D}^T \widehat{D})^{-1} (\widehat{D}^T x - \lambda \widehat{v}/2)$
   - Perform a discrete line search on the closed line segment from $\widehat{a}$ to $\widehat{a}_{\mathrm{new}}$
   - Check the objective value at $\widehat{a}_{\mathrm{new}}$ and all the points where any coefficient changes sign
   - Update $\widehat{a}$ and the corresponding entries in x to the point with the lowest objective value

Algorithm 2 continued

Remove the zero coefficients of the $\widehat{a}$ from U and update $v = \text{sign}(a)$

4. If the optimality conditions for nonzero coefficients is satisfied, $\frac{\partial \|x-Da\|^2}{\partial a_j} + \lambda \text{sign}(a_j) = 0, \quad \forall a_j \neq 0$

If the optimality conditions for zero coefficients is satisfied, $\left|\frac{\partial \|x-Da\|^2}{\partial a_j}\right| \leq \lambda, \quad \forall a_j = 0$

Set a as the optimal solution

Else

Go to step 2

Else

Go to step 3

End

The sparse coding with ELM is performed on the basis of the OP-ELM algorithm, which starts with a large SLFN by the original ELM algorithm and then ranks and eliminates the hidden nodes by the multi-response sparse regression algorithm (MRSR) and the LOO validation [38].

Algorithm 3 OP-ELM

1. Input the latest image patches X_0 and the corresponding sparse coding matrix A_0 from the natural images, the face image patches X, the activation function g and the maximum number of the hidden nodes $L_{\max}$, i.e., a large enough number denoting the number of kernels
2. Initialize the input weight W, the biases b with the random matrix

Set the number of the hidden nodes as $L_{\max}$

Calculate the hidden neuron output matrix $\mathbf{H} = g(W \times X_0 + b)$ and set the ith column of the hidden node matrix $\mathbf{H}$ as h_i

3. Rank the output of the hidden neuron h_i by their performance

Calculate the output weight $\boldsymbol{\beta} = \mathbf{H}^{\dagger} A_0$ by the Moore–Penrose pseudo-inverse

4. For $i = 1 : L_{\max}$

Get a submatrix of $\mathbf{H}$ that contains the columns from h_1 to h_i

Create a N-dimensional unit vector $I_{N\times 1} = [1 \quad 1 \quad \cdots \quad 1]'$

Construct the new matrix $H_i = [h_1, \cdots, h_i, I_{N\times 1}]_{N\times(i+1)}$

Introduce and compute the output weight matrix C_i that satisfies $C_i = H_i^{\dagger} A_0$

Calculate the error vector $\varepsilon_i = \frac{A_0 - H_i C_i}{1 - H_i (\mathbf{H}\mathbf{H}^T)^{-1} H_i}$ for each pixel in the image patch

End

Construct the generalization error matrix $\varepsilon = (\varepsilon_1', \varepsilon_2', \ldots, \varepsilon_{L_{\max}}')'$

Set the number of hidden nodes as $L(j) = \arg\min_i(\varepsilon(i,j)), j = 1, \ldots, p^2$

For $j = 1 : p^2$

Get a submatrix of $\mathbf{H}$ that contains the columns from h_1 to $h_{L(j)}$

Algorithm 3 continued

Create a N-dimensional unit vector $I_{N\times 1} = [1 \quad 1 \quad \cdots \quad 1]'$

Construct the new matrix $H_{L(j)} = [h_1, \ldots, h_{L(j)}, I_{N\times 1}]_{N\times(L(j)+1)}$

Calculate the output weight matrix $R_{L(j)} = (r_1, r_2, \ldots, r_{p^2})$ by the Moore–Penrose pseudo-inverse, $R_{L(j)} = H_{L(j)}^{\dagger} A_0$

End

5. Calculate again the hidden neuron output matrix $\mathbf{H} = g(W \times X + b)$
6. For $j = 1 : p^2$

Get a submatrix of $\mathbf{H}$ that contains the columns from h_1 to $h_{L(j)}$

Create a N'-dimensional unit vector $I_{N'\times 1} = [1 \quad 1 \quad \cdots \quad 1]'$

Construct the new matrix $H_{L(j)} = [h_1, \ldots, h_{L(j)}, I_{N'\times 1}]_{N'\times(L(j)+1)}$

Input the output weight matrix $R_{L(j)} = (r_1, r_2, \ldots, r_{p^2})$

Calculate the jth row of the sparse coding coefficient $A_j = H_{L(j)} r_j$ for face images

End

Set the sparse coding representation $A = [A_1, A_2, \ldots, A_{p^2}]_{N'\times p^2}$ as the output

Simulations and Result Analysis

Basis Function

In our simulation, the common feature was first extracted from the natural images by the efficient sparse coding algorithm. Figure 4 shows the original natural images. Multiple sets of common features with different dimensionality were taken here to evaluate the effect of overcompleteness for the recognition performance, i.e., the ratio between the dimensionality of the sparse coding confident $a_{0,i}$ and the image patch $x_{0,i}$. Figure 5a, b is the basis functions learned when $M = 512$, $N = 100$, and $p = 4$, $p = 8$, respectively, and Fig. 5c displays the 128 basis functions learned on natural image patches when $p = 8$.

Sparse Coding

Some classical face images were then taken to perform face recognition by means of the sparse coding with ELM, such as the ORL and PIE database. Here, we took the contributions in Shan and Cottrell [24] as the reference for our direct sparse coding. Figure 6 shows the difference of the square error in the sparse coding representation between the direct method and the proposed ELM method. As is shown, the difference of the two methods is quite small and it is possible to neglect the error and take the sparse coding extracted by ELM instead.

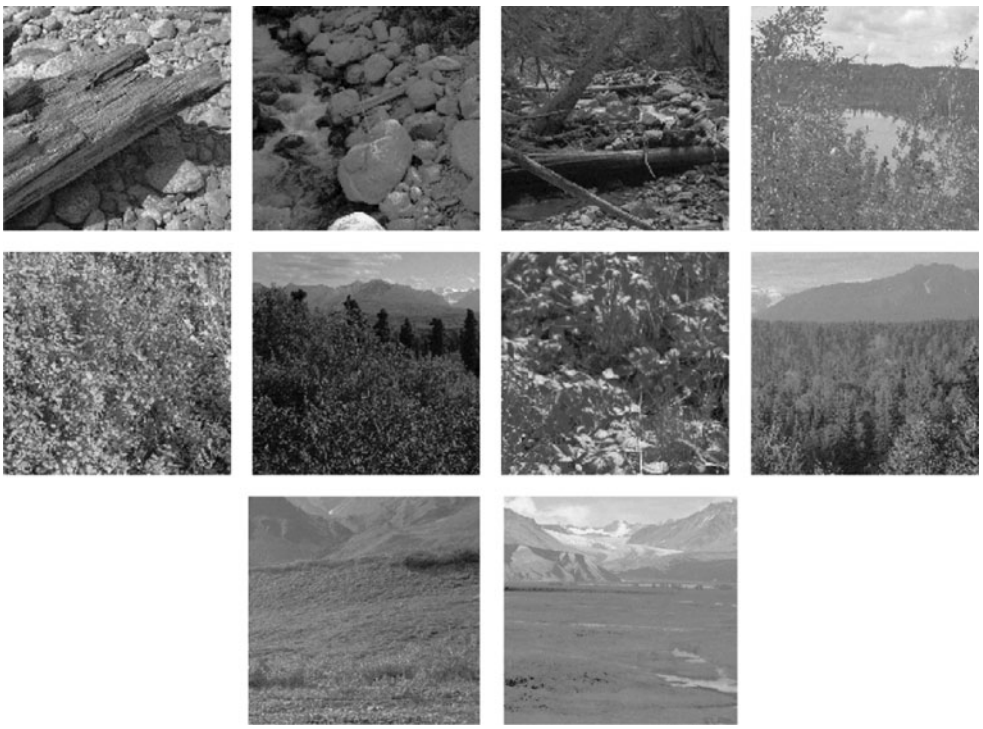

Fig. 4 Original natural images

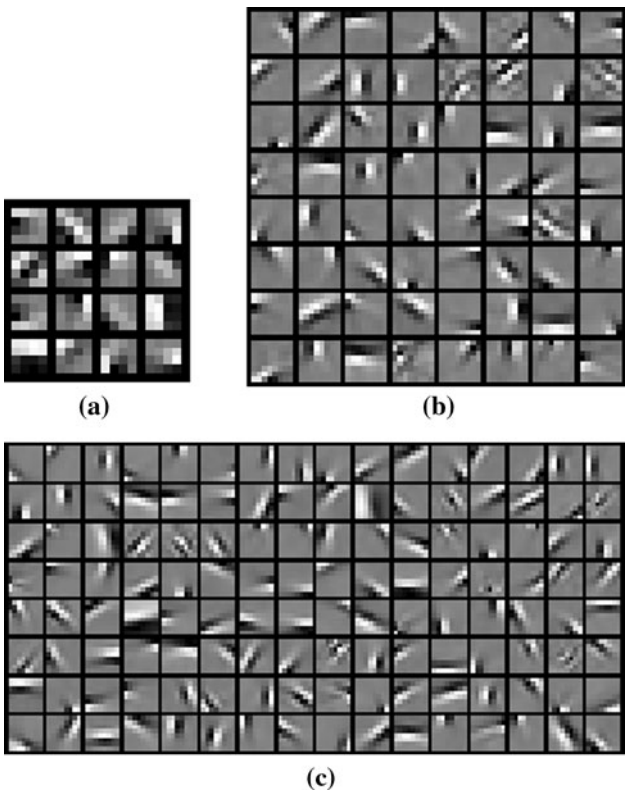

Fig. 5 Basis functions. **a** Basis function when $p = 4$. **b** Basis function when $p = 8$. **c** 128 basis functions learned on image patches with $p = 8$

The recovery strategy was adopted to further evaluate the performance of the two methods, by calculating $X = DA$. Figure 7 shows the example images recovered by the two methods, respectively. The peak signal-to-noise ratio (PSNR) as well as the running time of the sparse coding representation for one single image was taken here as some criterion for the performance evaluation of the two methods, and Table 1 lists the results for the face images

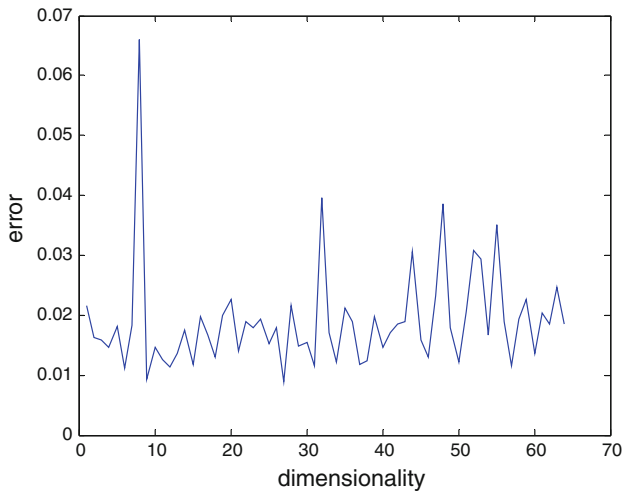

Fig. 6 Difference of the sparse coding representation

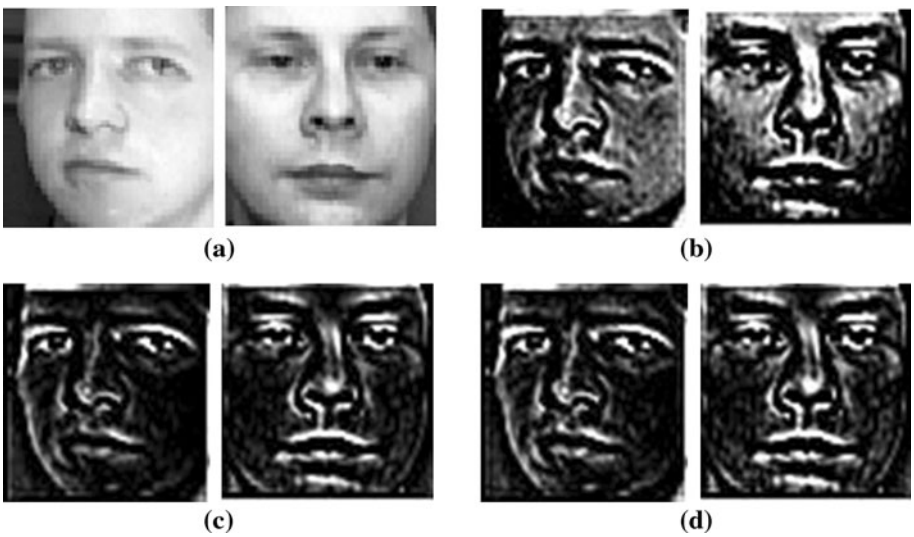

Fig. 7 Sparse coding recovery. **a** Original images. **b** Images by whitening transform. **c** Images recovered by sparse coding. **d** Images recovered by ELM

Table 1 Performance evaluation

Methods	Performance evaluation			
	PSNR (db)		Running time (s)	
	Image 1	Image 2	Image 1	Image 2
Sparse coding	52.7	54.7	2.02	2.11
ELM	53.34	55.8	0.57	0.75

shown in Fig. 7. Form the above evaluation, it was found that the ELM algorithm could recover the face images very well and sometimes tend to get a higher PSNR and less distortion than the direct sparse coding method. In addition, the running time of the ELM method was less than what the classical sparse coding cost, which guaranteed that the approach we proposed could outperform many recent techniques for face recognition at a much faster speed.

Face Recognition

ORL Database

ORL database is composed of 400 face images of 40 persons taken at different time, under different lighting conditions and with different facial expressions, with each person 10 images. Figure 8 shows the example images from ORL. The original face images were first scaled into 32×32. Taking $p = 8$ as an example, each face image was decomposed into all the $625 \times 8 \times 8$ image patches and represented by a 625×63 matrix after the PCA projection. The output of the first ELM learning algorithm was a 625×64 matrix for the sparse coding, which could be represented as a 40,000 dimensional vector for each image and feed the input of the next SLFN for face recognition.

During the recognition stage, we randomly selected $Q_1^{'} = 2, \ldots, 8$ images from each person as the training set,

Fig. 8 ORL face images

Table 2 Recognition rates on ORL database when $p = 8$

Methods	The size of the training set						
	2	3	4	5	6	7	8
Sparse coding	0.84	0.913	0.949	0.97	0.978	0.986	0.989
ELM	0.828	0.901	0.942	0.967	0.978	0.986	0.989

Table 3 Recognition rates on ORL database when $p = 4$

Methods	The size of the training set						
	2	3	4	5	6	7	8
Sparse coding	0.843	0.913	0.948	0.973	0.982	0.989	0.993
ELM	0.829	0.901	0.939	0.962	0.974	0.982	0.986

Table 4 Performance comparison of the classifiers in the recognition rates

Methods	The size of the training set						
	2	3	4	5	6	7	8
Softmax	0.83	0.903	0.942	0.968	0.975	0.986	0.990
M-SVM	0.831	0.903	0.943	0.968	0.978	0.986	0.990
ELM	0.828	0.901	0.942	0.967	0.978	0.986	0.989

Table 5 Performance comparison of the classifiers in the recognition speed

Method	Softmax	M-SVM	ELM
Time (s)	0.51	4.43	0.13

and the rest constitutes the test set. For each $Q_1^{'}$, 50 random splits were taken for the recognition performance. The principal components were extracted to make sure 95 % of the variance was captured, and with the increase of $Q_1^{'}$, the number of the principal comments ranged from 27 to 105. Tables 2 and 3 list the average recognition rates in percentage on ORL when $p = 4$ and $p = 8$. Here, the average recognition rates by the direct sparse coding in Table 2 are almost the same as the ones reported in Shan and Cottrell [24], and the performance of the proposed method was close to the direct sparse coding, but at a faster speed.

In order to evaluate the performance of the classifier, we also chose three different classifiers: one layer network with the softmax activation function, multi-linear SVM classifier, and multi-class ELM learning with sigmoid activation function adopted. When $p = 8$, the performance comparison is listed in Table 4. Table 5 also makes the comparison in the time duration for a single training and test when $Q_1^{'} = 2$. The ELM algorithm has an extraordinary advantage over the other classifiers in the execution speed.

PIE Database

In the PIE database, there are 41,368 face images from 68 individuals all together, and the images of each person were taken in 13 poses, under 43 illumination conditions and with 4 facial expressions. Figure 9 shows the example images from PIE.

We took those at 5 near frontal poses C05, C07, C09, C27, and C29 for our simulation, so 170 face images were selected for each individual. Every face image was first resized to 32 × 32. When randomly selecting $Q_1^{'} = 5$, 10, 20, 30, 50, 70, 90, 110 images for training and the rest for testing per person, the average recognition rates in percentage are listed in Table 6 with a comparison in the direct sparse coding, which shows that our method proposed had a better recognition performance at a higher speed, especially when the number of the face images increases greatly in practice.

Fig. 9 PIE face images

Table 6 Recognition rates on PIE database

Methods	The size of the training set							
	5	10	20	30	50	70	90	110
Sparse coding	79.09	91.85	97.05	97.94	98.64	99.04	99.22	99.41
ELM	80.16	92.98	97.54	98.23	98.84	99.10	99.29	99.50

Conclusion

In this paper, we have put forward a novel approach for face recognition, which have connected the ELM and sparse coding together for a faster solution. The idea of the fast face recognition is to learn the common feature hypothesis from the randomly collected universal images instead of directly from face images by means of ELM learning in SLFN, in case that the size of the face images is less than an optimum. The embedding basis function has been simulated with ELM to achieve the corresponding sparse coding representations from the face images at a higher speed. Some attempts have been made to develop the relevant mathematical criterion to capture the inherent distribution, dependence structure, and model selection in the ELM learning. The resulting sparse coding vectors of all the face images have further fed into the next SLFN for face recognition by ELM. The simulation results have shown the good performance comparable to the classical sparse coding on the ORL face data set, the PIE data set. Here, we have only taken face recognition as an example to evaluate the performance of our approach, and the proposed schemes could in fact also help other general applications in object recognition for the speed and performance improvements.

Acknowledgments This work was fully supported by the Natural Science Foundation of People's Republic of China (41176076) and the Natural Science Foundation of People's Republic of China (31202036).

References

1. Clark A. Mindware: an introduction to the philosophy of cognitive science. New York: Oxford University Press; 2001.
2. Underwood G. Cognitive processes in eye guidance: algorithms for attention in image processing. Cogn Comput. 2009;1(1):64–76.
3. Cambria E, Hussain A. Sentic album: content-, concept-, and context-based online personal photo management system. Cogn Comput. 2012;4(4):477–96.
4. Nian R, Ji GR, Zhao WC, Feng C. Probabilistic 3D object recognition from 2D invariant view sequence based on similarity. Neurocomputing. 2007;70(4–6):785–93.
5. Zhao W, Chellappa R, Phillips P, Rosenfeld A. Face recognition: a literature survey. ACM Comput Surv. 2003;35(4):399–458.
6. Turk M, Pentland A. Eigenfaces for recognition. J Cogn Neurosci. 1991;3:71–86.
7. Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell. 1997;19(7):711–20.
8. Cootes TF, Edwards GJ, Taylor CJ. Active appearance models. IEEE Trans Pattern Anal Mach Intell. 1998;23(6):681–5.
9. Brunelli R, Poggio T. Face recognition: features versus templates. IEEE Trans Pattern Anal Mach Intell. 1993;15(10):1042–52.
10. Heisele B, Serre T, Poggio T. A component-based framework for face detection and identification. Int J Comput Vision. 2007;74(2):167–81.
11. Zou J, Ji Q, Nagy G. A comparative study of local matching approach for face recognition. IEEE Trans Image Process. 2007;16(10):2617–28.
12. Wiskott L, Fellous JM, Kuiger N, von der Malsburg C. Face recognition by elastic bunch graph matching. IEEE Trans Pattern Anal Mach Intell. 1997;19(7):775–9.
13. Hubel DH, Wiesel TN. Receptive fields and functional architecture of monkey striate cortex. J Physiol. 1968;195(3):215–43.
14. Olshausen BA, Field DJ. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature. 1996;381(13):607–9.
15. Olshausen BA, Field DJ. Sparse coding with an over-complete basis set: a strategy employed by v1? Vis Res. 1997;37(23):3311–25.
16. Lee H, Battle A, Raina R, Ng AY. Efficient sparse coding algorithms. Adv Neural Inf Process Syst. 2006;801–8.
17. Lewicki MS, Sejnowski TJ. Learning overcomplete representations. Neural Comput. 2000;12(2):337–65.
18. Johnson JS, Olshausen BA. Timecourse of neural signatures of object recognition. J Vis. 2003;3(7):499–512.
19. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y. Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell. 2009;31(2):210–27.
20. Huang JZ, Huang XL, Metaxas D. Simultaneous image transformation and sparse representation recovery. CVPR. 2008;1–8.
21. Wagner A, Wright J, Ganesh A, Zhou ZH, Ma Y. Towards a practical face recognition system: robust registration and illumination by sparse representation. CVPR. 2009;597–604.
22. Yang M, Zhang L. Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary. ECCV. 2010;448–61.
23. Shan HH, Zhang LY, Cottrell GW. Recursive ICA. Adv Neural Inf Process Syst. 2006;1273–80.
24. Shan HH, Cottrell GW. Looking around the backyard helps to recognize faces and digits. CVPR. 2008;1–8.
25. Huang GB, Zhu Q, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006;70:489–501.
26. Huang GB, Wang DH, Lan Y. Extreme learning machines: a survey. Int J Mach Learn Cybern. 2011;2:107–22.
27. Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw. 2006;17(4):879–92.
28. Huang GB, Chen L. Convex incremental extreme learning machine. Neurocomputing. 2007;70:3056–62.
29. Huang GB, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing. 2008;71:3460–8.

30. Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multi-class classification. IEEE Trans Syst Man Cybern. 2012;42(2):513–29.
31. Frénay B, Verleysen M. Using SVMs with randomized feature spaces: an extreme learning approach. In: Proceedings of the 18th European symposium on artificial neural networks (ESANN), Bruges, Belgium, 28–30 April 2010; 2010. p. 315–20.
32. Frénay B, Verleysen M. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing. 2011;74(16):2526–31.
33. Huang GB, Li M, Chen L, Siew C-K CK. Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing. 2008;71:576–83.
34. Liang N, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate on-line sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw. 2006;17(6):1411–23.
35. Rong HJ, Huang GB, Sundararajan N, Saratchandran P. Online sequential fuzzy extreme learning machine for function approximation and classification problems. IEEE Trans Syst Man Cybern. 2009;39(4):1067–72.
36. Sun Y, Yuan Y, Wang G. An OS-ELM based distributed ensemble classification framework in P2P networks. Neurocomputing. 2011;74:2438–43.
37. Rong H, Ong YS, Tan AH, Zhu Z. A fast pruned extreme learning machine for classification problem. Neurocomputing. 2008;72:359–66.
38. Miche Y, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A. OP-ELM: optimally-pruned extreme learning machine. IEEE Trans Neural Netw. 2010;21(1):158–62.
39. Decherchi S, Gastaldo P, Zunino R, Cambria E, Redi J. Circular-ELM for the reduced-reference assessment of perceived image quality. Neurocomputing. 2013;102:78–89.
40. van Heeswijk M, Miche Y, Lindh-Knuutila T, Hilbers PA, Honkela T, Oja E, Lendasse A. Adaptive ensemble models of extreme learning machines for time series prediction. Lect Notes Comput Sci. 2009;5769:305–14.
41. van Heeswijk M, Miche Y, Oja E, Lendasse A. Gpu accelerated and parallelized ELM ensembles for large-scale regression. Neurocomputing. 2011;74:2430–7.
42. Man ZH, Lee K, Wang DH, Cao ZW, Miao CY. A new robust training algorithm for a class of single-hidden layer feedforward neural networks. Neurocomputing. 2011;74:2491–501.
43. Man ZH, Lee K, Wang DH, Cao ZW, Miao CY. A modified ELM algorithm for single-hidden layer feedforward neural networks with linear nodes. In: 2011 6th IEEE conference on industrial electronics and applications; 2011. p. 2524–9.
44. Cheng C, Tay WP, Huang GB. Extreme learning machines for intrusion detection. In: International joint conference on neural networks (IJCNN); 2012. p. 1–8.
45. Nian R, He B, Lendasse A. 3D object recognition based on a geometrical topology model and extreme learning machine. Neural Comput Appl. 2012;22(3–4):427–33.
46. Wang GR, Zhao Y, Wang D. A protein secondary structure prediction framework based on the extreme learning machine. Neurocomputing. 2008;72:262–8.
47. Bartlett PL. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Trans Inf Theory. 1998;44(2):525–36.
48. Hansen LK, Salamon P. Neural network ensemble. IEEE Trans Pattern Anal Mach Intell. 1990;12(10):993–1001.