Multidim Syst Sign Process (2019) 30:1149–1169
https://doi.org/10.1007/s11045-018-0598-9

Local receptive fields based extreme learning machine with hybrid filter kernels for image classification

Bo He[1] · Yan Song[1] · Yuemei Zhu[1] · Qixin Sha[1] · Yue Shen[1] · Tianhong Yan[2] · Rui Nian[1] · Amaury Lendasse[3,4]

Received: 25 November 2017 / Revised: 20 May 2018 / Accepted: 2 June 2018 /
Published online: 16 June 2018

Abstract In this paper, an innovative method called extreme learning machine with hybrid local receptive fields (ELM-HLRF) is presented for image classification. In this method, filters generated by Gabor functions and the randomly generated convolution filters are incorporated into the convolution filter kernels of local receptive fields based extreme learning machine (ELM-LRF). Extreme learning machine (ELM) is derived from single hidden layer feed-forward neural networks, and the parameters of its hidden layer can be generated randomly. As locally connected ELM, ELM-LRF directly processes information with strong correlations such as images and speech. In this paper, two main contributions are proposed to improve the classification performance of ELM-LRF. First, the Gabor functions are used as one kind of convolution filter kernels of ELM-HLRF to execute image classification. Second, we use a data augmentation method to preprocess training images to avoid overfitting. Experiments on the Outex texture dataset, the Yale face dataset, the ORL face database and the NORB dataset demonstrate that ELM-HLRF outperforms ELM-LRF, ELM and support vector machine in classification accuracy, and the presented data augmentation method improves the classification performance.

Keywords Image classification · Local receptive fields · Extreme learning machine · Gabor filters · Data augmentation

Bo He and Yan Song have contributed equally to this work.

✉ Bo He
bhe@ouc.edu.cn

1 School of Information Science and Engineering, Ocean University of China, 238 Songling Road, Qingdao 266100, China

2 School of Mechanical and Electrical Engineering, China Jiliang University, 258 Xueyuan Street, Xiasha High-Edu Park, Hangzhou 310018, China

3 Department of Mechanical and Industrial Engineering and the Iowa Informatics Initiative, The University of Iowa, Iowa City, IA 52242-1527, USA

4 Arcada University of Applied Sciences, Helsinki, Finland

1 Introduction

Machine learning has been extensively studied in image classification and recognition nowadays. In some methods, features extraction and selection play an important role before applying neural networks to execute classification. Whether the features can represent the characteristics of images matters a lot and determines the classification accuracy. Local binary patterns (LBPs) (Pietikäinen 2010) and gray level co-occurrence matrix (GLCM) (Haralick et al. 1973) are features widely used for texture classification. On the other hand, researchers have utilized neural networks with local receptive field, such as convolutional neural network (CNN) (LeCun et al. 1995), to process images directly without extra feature extraction step. In CNN, the structure formed by convolutional layer and pooling layer performs like feature extraction procedure.

Extreme learning machine (ELM) was proposed by Huang et al. (2006, 2012) and performed well in both regression and classification. ELM is derived from single-hidden layer feed-forward neural networks (SLFNs) which consists of one hidden layer and one output layer. The advantages of ELM include that the hidden layer parameters of ELM can be generated randomly and it learns faster while keeping superior performance. ELM has been used in image classification and recognition combined with other algorithms or various kinds of image features. Li et al. (2015) employed ELM as the classifier with image features extracted by LBPs, which presented better performance than the state-of-art methods. Meanwhile, ELM and graph-based optimization methods were fused to boost remote sensing image classification (Bencherif et al. 2015). And Zeng et al. 2017 presented a traffic sign recognition method in which CNN was used to extract features from images and ELM was also applied as the classifier.

To utilize ELM to process images directly, Huang et al. (2015) proposed local receptive field based extreme learning machine (ELM-LRF). ELM-LRF outperforms CNN on the NORB dataset (LeCun et al. 2004a) in classification accuracy and time consumption. Huang et al. (2017) proposed a modified ELM-LRF to perform texture image classification. The modified method (Huang et al. 2017) employed multi-scale convolution kernels in ELM-LRF (ELM-MSLRF) and it could learn texture information of different scales. ELM-MSLRF is superior to ELM-LRF according to the experimental results.

The Gabor filters (Fogel and Sagi 1989) have been successfully employed in object detection (Jain et al. 1997), image segmentation (Jain and Farrokhnia 1991) and classification (Rajadell et al. 2013) and edge detection (Mehrotra et al. 1992) for more than two decades. By providing information with different scales and orientations, Gabor filters perform like human beings and are often used for texture representation and description. In practical applications, Gabor filters can extract the relevant features in different scales and orientations in the frequency domain. Recently, Gabor filters also have been used as the convolution kernels of CNN (GCNN) to carry out improved speech recognition (Chang and Morgan 2014).

On the other hand, data augmentation has been widely used in neural network to prevent overfitting of small dataset (Cui et al. 2015; Krizhevsky et al. 2012). In these studies, label-preserving transformations were used to augment training data. Cui et al. (2015) proposed two data augmentation methods to deal with data sparsity for both deep neural networks (DNNs) and CNN. The proposed augmentation methods could increase the variations of training data. Krizhevsky et al. (2012) introduced two data augmentation forms to reduce overfitting. The first form was carried out by doing image translation and horizontal reflections, and the second one included altering the intensities of the RGB channels in training images.

Motivated by the aforementioned research, two main improvements are introduced in this paper. First, to improve the performance of ELM-LRF in image classification, we use Gabor functions as one kind of convolution kernel filters. The Gabor functions can provide more image information by using filters with different scales and orientations. The proposed method, extreme learning machine with hybrid local receptive fields (ELM-HLRF), can provide maps generated by Gabor filters and randomly generated convolution filters in the convolutional layer. Second, we propose a data augmentation method using label-preserving transformations to improve classification performance. This data augmentation method uses Gaussian blur to preprocess training images. Then the blurred images and the original training images will be incorporated as augmented data to train classifiers. We evaluate the proposed methods on these datasets: the Outex dataset (Ojala et al. 2002), the Yale face database (Georghiades et al. 1997), the ORL face database (Samaria and Harter 1994) and the NORB dataset (LeCun et al. 2004b). Experimental results demonstrate that: first, ELM-HLRF performs better than ELM-LRF, ELM and support vector machine (SVM) (Cortes and Vapnik 1995) in classification accuracy; second, the proposed data augmentation method can improve classification performance.

The rest of this paper begins with related works in Sect. 2. Section 3 gives a detailed description of the proposed methods. Section 4 reports and discusses the experimental results. Finally Sect. 5 concludes.

2 Related works

2.1 Gabor filters in image processing

The two-dimensional discrete Gabor function can be written as follows

$$\phi(x, y) = \frac{1}{2\pi\delta_x\delta_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\delta_x^2}+\frac{y^2}{\delta_y^2}\right)\right]\exp(2\pi j W x), \tag{1}$$

where W denotes the radial frequency of Gabor wavelet, δ_x and δ_y are parameters of Gaussian envelope along the x-axis and y-axis respectively.

The Gabor filter with frequency W and orientation θ by coordinate rotation can be given by

$$\phi'(x, y) = \frac{1}{2\pi\delta_x\delta_y}\exp\left[-\frac{1}{2}\left(\frac{x'^2}{\delta_x^2}+\frac{y'^2}{\delta_y^2}\right)\right]\exp(2\pi j W x'), \tag{2}$$

where $\phi'(x, y)$ is the two-dimensional discrete Gabor function when $x' = \alpha^{-s}(x \cdot \cos\theta_l + y \cdot \sin\theta_l)$, $y' = \alpha^{-s}(-x \cdot \sin\theta_l + y \cdot \cos\theta_l)$, s is the scale and l is the orientation; * represents transpose conjugate; $s = 1, 2, \ldots, p$; $l = 1, 2, \ldots, q$; α is the scale factor and $\alpha > 1$. The (x, y) denotes initial coordinate, while (x', y') denotes the transformed coordinate. The symbols p and q are the number of scales and orientations of Gabor filters respectively. The functions $\phi(x, y)$ and $\phi'(x, y)$ have the following relationship:

$$\phi(x, y) = \alpha^{-s}\phi'(x, y). \tag{3}$$

The Fourier transform of $\phi(x, y)$ is

$$\Phi(u, v) = \exp\left\{\frac{1}{2}\left[\frac{(u-W)^2}{\delta_u^2}+\frac{v^2}{\delta_v}\right]\right\}, \tag{4}$$

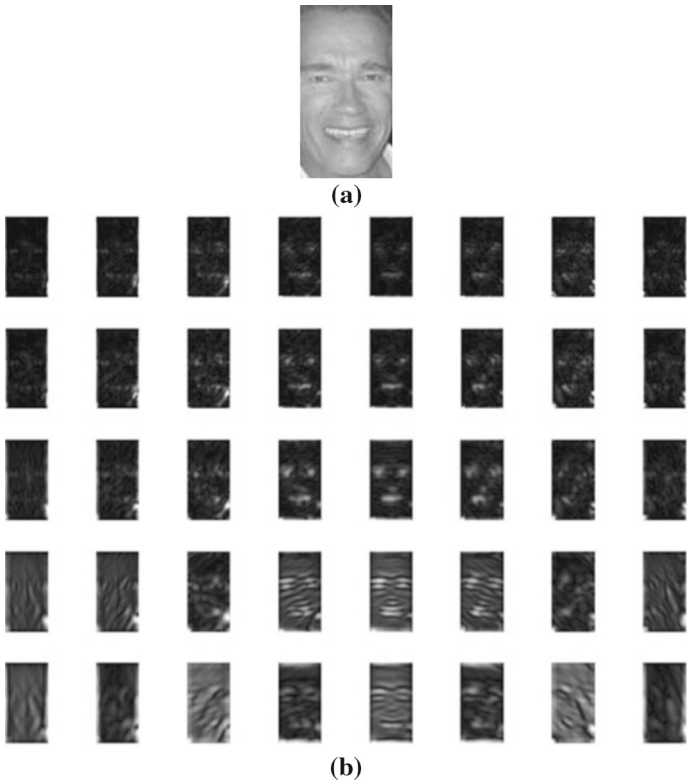


Fig. 1 **b** Gabor filtered results of **a** image. In **b**, each row shares the same scale and each column shares the same orientation

where $\delta_u = 1/2\pi\delta_x$ and $\delta_v = 1/2\pi\delta_y$. Let $I(x, y)$ be the input image, $I(x, y)$ filtered by $\phi'(x, y)$ can be written as

$$F(x, y) = \sum_{x_1} \sum_{y_1} I(x_1, y_1)\phi'^*(x - x_1, y - y_1), \tag{5}$$

where $F(x, y)$ is the filter response. The mean and standard deviation of the magnitude of $F(x, y)$ can be used as features (Manjunath and Ma 1996) of image tiles to perform texture classification. The mean and standard derivative of $F(x, y)$ are calculated as

$$\mu_{s,l} = \sum_{x} \sum_{y} |F(x, y)|, \tag{6}$$

$$\delta_{s,l} = \sqrt{\sum_{x} \sum_{y} (|F(x, y)| - \mu_{s,l})^2}. \tag{7}$$

The feature vector for $I(x, y)$ is represented as $[\mu_{1,1}, \delta_{1,1}, \mu_{1,2}, \delta_{1,2}, \ldots, \mu_{p,q}, \delta_{p,q}]$.

Figure 1 presents an instance of Gabor filtered face image. In this instance, we set $p = 5$ and $q = 8$, so there are 40 Gabor filtered results.

2.2 Brief review of ELM

ELM is proposed by Huang et al. (2006, 2012). When compared with traditional neural networks, ELM has faster learning speed and higher accuracy. The weights and biases in hidden layer of ELM can be assigned randomly. The flowchart of ELM is shown in Fig. 2. Let (X_j, t_j) $(j = 1, 2, \ldots, N)$ be the N input samples of SLFNs, where $X_j = [x_{j1}, x_{j2}, \ldots, x_{jn}]^T \in R^n$ denotes input feature vector and t_j denotes target value of X_j . The value domain of t_j is $\{1, 2, \ldots, m\}$. The SLFNs with L hidden nodes can be written as follows

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot X_j + b_i) = o_j, \quad j = 1, \ldots, N, \tag{8}$$

where $g(\cdot)$ denotes the activation function, $W_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,n}]$ denotes the input weights, β_i denotes the output weight and b_i denotes the bias of the ith hidden layer unit. Function $W_i \cdot X_j$ denotes the inner product of W_i and X_j.

The goal of SLFNs is to minimize the output error, that is to say, the output of SLFNS o_j and the target output t_j should satisfy the following equation:

$$\sum_{j=1}^{N} \left\| o_j - t_j \right\| = 0. \tag{9}$$

For the training dataset, we have the following assumption:

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot X_j + b_i) = t_j, \quad j = 1, \ldots, N, \tag{10}$$

$$H\beta = T, \tag{11}$$

where H is the output matrix of the hidden layer, β is the output weights and $T = [t_1, t_2, \ldots, t_N]$ is the target outputs of all inputs. Equation 11 can be written as

$$\begin{aligned} &H(W_1, \cdots, W_L, b_1, \cdots, b_L, X_1, \ldots, X_L) \\ &= \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & \cdots & g(W_L \cdot X_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(W_1 \cdot X_N + b_1) & \cdots & g(W_L \cdot X_N + b_L) \end{bmatrix}_{N \times L}, \end{aligned} \tag{12}$$

where

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1}, T = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{N \times 1}. \tag{13}$$

After training, we get $\hat{W}_i$, $\hat{b}_i$ and $\hat{\beta}_i$ which satisfy the following equation

$$\left\| H(\hat{W}_i, \hat{b}_i)\hat{\beta}_i - T \right\| = \min_{W,b,\beta} \left\| H(W_i, b_i)\beta_i - T \right\|, \tag{14}$$

where $i = 1, \ldots, L$. Equation (14) also means to minimize loss function also means to minimize loss function

$$E = \sum_{j=1}^{N} \left(\sum_{i=1}^{L} \beta_i g(W_i \cdot X_j + b_i) - t_j \right)^2. \tag{15}$$

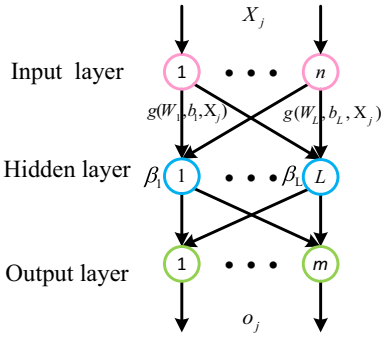


Fig. 2 The flowchart of ELM

β can be calculated as follows

$$\hat{\beta} = H^{\dagger} T, \tag{16}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse of H.

2.3 Brief review of ELM-LRF

In ELM-LRF, the connection between the input layer and one node of hidden layer is generated according to a continuous probability distribution. These random connections constitute local receptive fields. ELM-LRF consists of four layers: the hidden layer, the pooling layer, the full-connected layer and the output layer.

In the hidden layer, the convolution kernel a_i $(i = 1, 2, \ldots, k')$ is randomly generated. Assume that the initial input weights are $\hat{A}^{init}$, the size of each input weight is $r \times r$ and the size of each input image is $d \times d$. So the size of each feature map is $(d - r + 1) \times (d - r + 1)$. Then

$$\begin{aligned} &\hat{A}^{init} \in R^{r^2 \times k'}, \hat{A}^{init} = [\hat{a}_1^{init}, \hat{a}_2^{init}, \cdots, \hat{a}_{k'}^{init}], \\ &\hat{a}_i^{init} \in R^{r^2}, i = 1, \ldots, k', \end{aligned} \tag{17}$$

where $\hat{A}^{init}$ is orthogonalised using singular value decomposition (SVD). The orthogonalised input weights are $\hat{A}$ and $\hat{A} = [\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{k'}]$. Each column of $\hat{A}$ is the orthogonal basis of $\hat{A}^{\text{init}}$. If $r^2 < k'$, $\hat{A}^{init}$ should be transposed at first and then orthogonalised and transposed again at last. The convolution weight of the ith feature map is $a_i \in \mathrm{R}^{r \times r}$ and it is aligned of $\hat{a}_i$ by column. The convolution result of node (x_1, x_2) at the ith feature map is $c_{x_1, x_2, i}$:

$$\begin{aligned} c_{x_1, x_2, i} &= \sum_{m_1=1}^{r} \sum_{m_2=1}^{r} (I_{x_1+m_1-1, x_2+m_2-1} \cdot a_{m_1, m_2, i}) \\ x_1, x_2 &= 1, \ldots, (d - r + 1), \end{aligned} \tag{18}$$

where $I_{x_1+m_1-1, x_2+m_2-1}$ is the pixel value of input image I at location $(x_1 + m_1 - 1, x_2 + m_2 - 1)$.

In the pooling layer, pooling size e denotes the distance between the center and the edge of pooling. And the size of the pooled maps are the same as the feature maps $((d - r + 1) \times (d - r + 1))$. The symbol $c_{x_1, x_2, i}$ is node (x_1, x_2) of the ith feature map; $h_{p_1, p_2, i}$ is node (p_1, p_2) of the ith pooled map; $i = 1, 2, \ldots, k'$. And $h_{p_1, p_2, i}$ is obtained using

$$\begin{aligned} h_{p_1, p_2, i} &= \sqrt{\sum_{x_1=p_1-e}^{p_1+e} \sum_{x_2=p_2-e}^{p_2+e} c_{x_1, x_2, i}^2} \\ p_1, p_2 &= 1, \ldots, (d - r + 1). \end{aligned} \tag{19}$$

If (x_1, x_2) is out of bound, $c_{x_1,x_2,i} = 0$.

In the full-connected layer, each pooling map is merged into a row vector. The size of each pooling map is $(d - r + 1) \times (d - r + 1)$. If there are N input images, the matrix $H' \in \mathrm{R}^{N \times [k' \cdot (d-r+1)^2]}$ is calculated as

$$H' = \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{1,2} & \cdots & \hat{h}_{1,k' \cdot (d-r+1)^2} \\ \hat{h}_{2,1} & \hat{h}_{2,2} & \cdots & \hat{h}_{2,k' \cdot (d-r+1)^2} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{h}_{N,1} & \hat{h}_{N,2} & \cdots & \hat{h}_{N,(k') \cdot (d-r+1)^2} \end{bmatrix}_{N \times [k' \cdot (d-r+1)^2]}, \tag{20}$$

where $\hat{h}_{i,j} = g(W_j \cdot I_i + b_j)$, I_i is the ith $(1 \le i \le N)$ input image and $1 \le j \le k' \cdot (d - r + 1)^2$. The output weight β of ELM-LRF is calculated as (Huang et al. 2012, 2015)

$$\beta = {H'}^T \left(\frac{1}{C} + H' {H'}^T \right)^{-1} T \quad \text{if } N \le k' \cdot (d - r + 1)^2, \tag{21}$$

$$\beta = \left(\frac{1}{C} + {H'}^T H' \right)^{-1} {H'}^T T \quad \text{if } N > k' \cdot (d - r + 1)^2. \tag{22}$$

3 Methods

Section 3.1 introduces the proposed method ELM-HLRF, and Sect. 3.2 presents the data augmentation method.

3.1 Local receptive field based extreme learning machine with hybrid filter kernels

In this paper, we propose an innovative neural network which uses Gabor filters and randomly generated convolution kernels in the convolutional layer. The proposed method is called hybrid local receptive field based extreme learning machine (ELM-HLRF). In this modified topology, randomly generated convolution kernels used in ELM-LRF and Gabor filters of different scales and orientations are combined to process input images.

In image processing, with different combinations of the scale and orientation parameters, Gabor filters are employed to detect contours of various scales and orientations. Therefore the scales and orientations of Gabor filters are important parameters, and these parameters need to be chosen to get optimal training results in ELM-HLRF.

The flowchart of ELM-HLRF is shown in Fig. 3. In Fig. 3, $G_{i_1} \in \mathrm{R}^{r \times r}$ $(i_1 = 1, \ldots, k_1)$ is the convolution kernel provided by Gabor filters, $k_1 = p \cdot q$ and

$$G_{i_1} = \phi'_{i_1}(x, y), \quad i_1 = 1, \ldots, k_1. \tag{23}$$

The convolution result when $\phi'_{i_1}(x, y)$ is applied to input image I is

$$F_{i_1}(x, y) = \sum_{x_1} \sum_{y_1} I(x_1, y_1) \phi'^{*}_{i_1}(x - x_1, y - y_1). \tag{24}$$

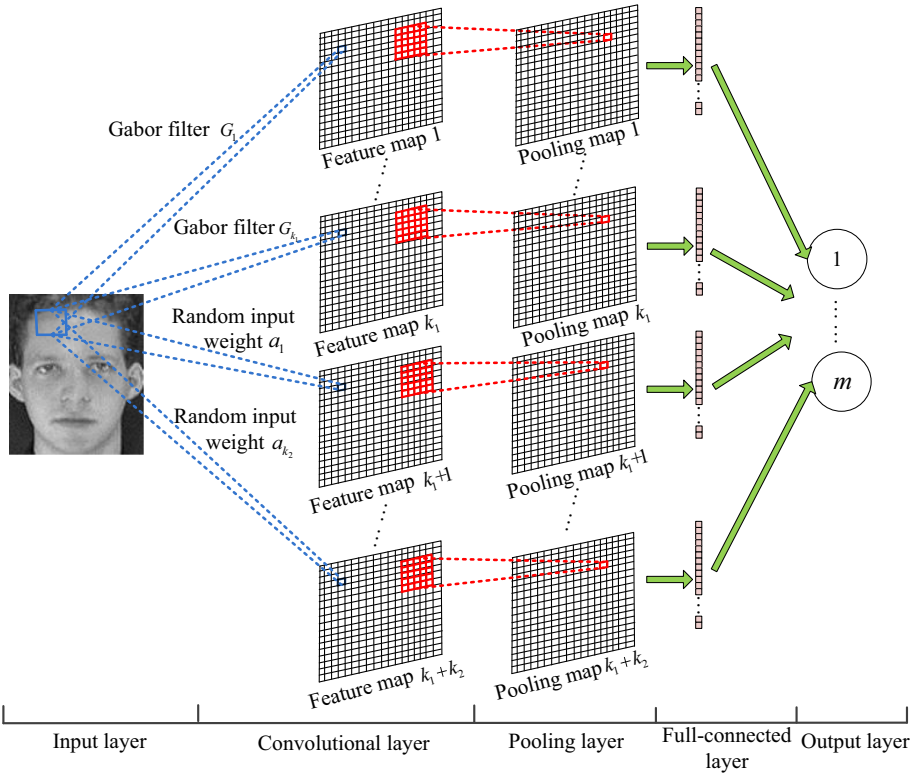


Fig. 3 The flowchart of ELM-HLRF

The other kind of convolution kernel a'_{i_2} $(i_2 = 1, 2, \ldots, k_2)$ is randomly generated. According to (17), the initial input weights are $\hat{A'}^{init}$, and

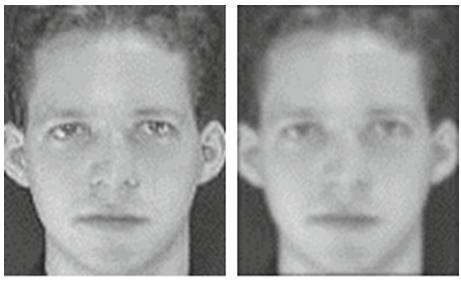
$$\begin{aligned} &\hat{A'}^{init} \in R^{r^2 \times k_2}, \hat{A'}^{init} = \left[\hat{a'}_1^{init}, \hat{a'}_2^{init}, \ldots, \hat{a'}_{k_2}^{init}\right], \\ &\hat{a'}_{i_2}^{init} \in R^{r^2}, \quad i_2 = 1, \ldots, k_2. \end{aligned} \tag{25}$$

The orthogonalised result of $\hat{A'}^{init}$ is $\hat{A'} = [\hat{a'}_1, \hat{a'}_2, \ldots, \hat{a'}_{k_2}]$. The convolution result of node (x_1, x_2) at the i_2th feature map is c'_{x_1,x_2,i_2}, which is obtained using (18).

The convolution weights of the convolutional layer of ELM-HLRF consist of $G_{i_1} \in \mathrm{R}^{r \times r}$ $(i_1 = 1, \ldots, k_1)$ and $\mathrm{a'}_{i_2} \in \mathrm{R}^{r \times r}$ $(i_2 = 1, \ldots, k_2)$. The pooling map $h'_{p_1,p_2,i}$ $(i = 1, 2, \ldots, (k_1 + k_2))$ can be calculated using (19).

As presented in Sect. 2.3, in the full-connected layer, the matrix $\hat{H} \in \mathrm{R}^{N \times [(k_1+k_2)\cdot(d-r+1)^2]}$ can be calculated as

$$\hat{H} = \begin{bmatrix} \hat{h}_{1,1} & \hat{h}_{1,2} & \cdots & \hat{h}_{1,(k_1+k_2)\cdot(d-r+1)^2} \\ \hat{h}_{2,1} & \hat{h}_{2,2} & \cdots & \hat{h}_{2,(k_1+k_2)\cdot(d-r+1)^2} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{h}_{N,1} & \hat{h}_{N,2} & \cdots & \hat{h}_{N,(k_1+k_2)\cdot(d-r+1)^2} \end{bmatrix}_{N \times [(k_1+k_2)\cdot(d-r+1)^2]}. \tag{26}$$



(a) **(b)**

Fig. 4 **b** Gaussian blurred result of the **a** original face image

The output weight ξ of ELM-HLRF is calculated as

$$\xi = \hat{H}^T \left(\frac{1}{C} + \hat{H}\hat{H}^T\right)^{-1} T \quad \text{if } N \leq (k_1 + k_2) \cdot (d - r + 1)^2, \tag{27}$$

$$\xi = \left(\frac{1}{C} + \hat{H}^T\hat{H}\right)^{-1} \hat{H}^T T \quad \text{if } N > (k_1 + k_2) \cdot (d - r + 1)^2. \tag{28}$$

3.2 Data augmentation

In supervised machine learning, we need sufficient data to train the neural networks to obtain robust model and to avoid overfitting. Data augmentation is the commonly used method which can enlarge datasets by using label-preserving transformations. The related data augmentation methods include color jittering, PCA jittering, random scale transformation, random crop, horizontal flip, vertical flip, translation transform, rotation, reflection, affine transformation, Gaussian noise and blurring, etc.

In this paper, we carry out a data augmentation method by altering the pixel intensities of training images. A 5×5 Gaussian blur function with standard deviation 1 is used to process each training image. Each blurred image and its corresponding training image share the same label. Then we train the network using the blurred images and the original training images. Figure 4 gives an example of Gaussian blurred face image. Trained with the augmented datasets, the learning machine will be more robust to blur details.

4 Experimental results and analysis

In this paper, we use five datasets to evaluate the performance of ELM-HLRF. Details about the datasets, the parameters setup, the experimental results and the discussions are introduced in this section.

4.1 Datasets for performance evaluation

In this paper, we use five datasets to evaluate the performance of ELM-HLRF: the Outex_TC_00000, the Outex_TC_00012, the Yale face database, the ORL face database and the NORB dataset.

We use two different databases of Outex: Outex_TC_00000 and Outex_TC_00012. The Outex_TC_00000 consists of 8832 images of 24 different textures. Each texture has 368

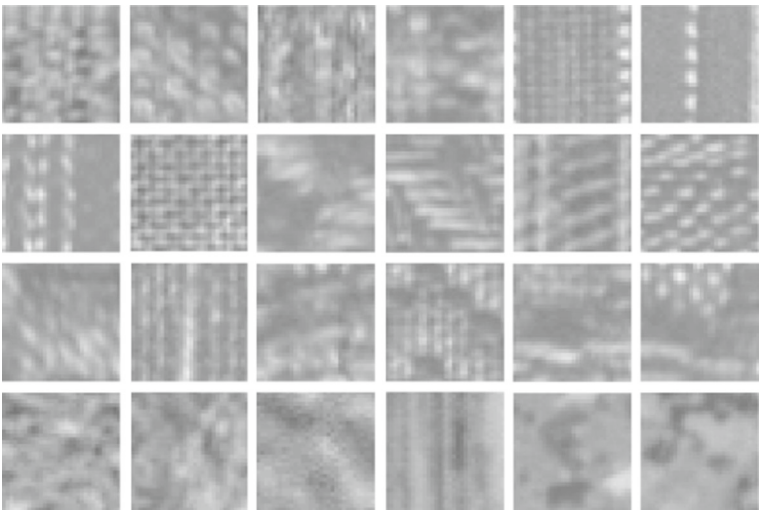


Fig. 5 24 different textures of the Outex_TC_00000 dataset

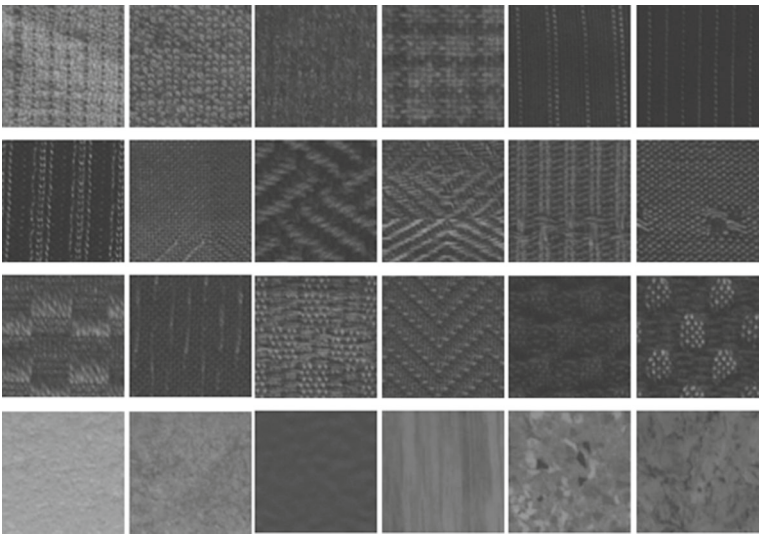


Fig. 6 24 different textures of the Outex_TC_00012 dataset

images, 184 for training and 184 for testing. We choose 099 folder in Outex_TC_00000 to do training and testing in our experiment. The other Outex database we use in this paper is Outex_TC_00012, which contains 1440 images of 24 different textures. Each texture has 60 samples, 20 of which for training and 40 of which for testing. These texture images are recorded under different illuminations. Images of Outex_TC_00012 are with the resolution of 128 by 128. We resize them into size of 32 by 32. Figures 5 and 6 give some texture image samples of Outex_TC_00000 and Outex_TC_00012 respectively.

The Yale face database contains 165 grayscale images of 15 individuals. There are 11 images for each face with different facial expressions such as center-light, glasses, happy,

Fig. 7 Images of the Yale face database

left-light, no glasses, normal, right-light, sad, sleepy, surprised and wink. Each image is 100 by 100 pixels in size. Images of Yale face database are also resized into a size of 32 by 32 pixels to speed up experiments in this paper. Figure 7 shows the face image samples of Yale face database.

The ORL face database (Samaria and Harter 1994) consists of 400 images, 10 for each of 40 distinct subjects. For some subjects, the images were taken at different times, lighting, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. The size of each image is 92 by 112 pixels. Images of the ORL face database are also resized into the size of 32 by 32 in this paper. Figure 8 are the face image samples of the ORL face database.

The NORB dataset is a benchmark for object recognition (LeCun et al. 2004c). It contains images of 50 toys belonging to 5 generic categories: four-legged animals, human figures, airplanes, trucks, and cars. The objects were imaged by two cameras under 6 lighting conditions, 9 elevations (30°–70° every 5°), and 18 azimuths (0–340 every 20°). And there are 48,600 images ($50 * 6 * 9 * 18$) in the NORB dataset, half of which (ie 24,300 images) are used for training and the rest (ie 24,300 images) for testing. We downsize them to 32×32 in the experiments.

4.2 Parameters setup

The experiments are carried out on five datasets using the Matlab software on a windows 7 64 bit system with Intel(R) Core(TM) i5-4210U CPU and 64 GB RAM.

In this paper, the number of the hidden nodes of ELM ranges from 100 to 2000. For each experiment, we will find the parameters which generate the optimal classification results on each dataset. Besides, six parameters have direct effect on classification accuracy: the number of the Gabor filter scales p, the number of the Gabor filter orientations q, the number of the convolution filters of ELM-HLRF k_2, the number of the convolution filters of ELM-LRF k', the convolution size r, the pooling size e and the regularization parameter C. The value of C can be $\{0.01, 0.1, 1, 10, 100\}$. The p ranges from 1 to 5 with stride 1, q is equal to 4 or 8,

Fig. 8 Image samples of the ORL face database

Table 1 The parameters of ELM-HLRF on dataset Outex_TC_00000 and Outex_TC_00012

k_2	r	e	C	p	q
48	9	8	0.01	1, 2, 3, 4, 5	4, 8

k' and k_2 ranges from 4 to 80 with stride 4. The convolution size r ranges from 4 to 9 and the pooling size e ranges from 3 to 8.

4.3 Results and analysis

4.3.1 Performance evaluation on dataset Outex_TC_00012

Figure 9 shows the relationship between the classification accuracy, the pooling size e and the convolution size r of ELM-LRF when the number of the convolution filters k_2 is set to 48. We can see from Fig. 9 that when $e = 8$ and $r = 9$ we have the highest classification accuracy. For Outex datasets Outex_TC_00000 and Outex_TC_00012, the parameters of ELM-HLRF are listed in Table 1.

Table 2 presents the classification accuracy when ELM-HLRF and ELM-LRF are applied to dataset Outex_TC_00012. It can be seen from Table 2 that the proposed method ELM-HLRF can improve classification accuracy by providing Gabor filtered maps in the convolutional layer. To evaluate the quality of Gabor filters as convolution kernels, we compare the classification accuracy of ELM-HLRF with that of ELM-LRF, ELM and SVM. Table 3 shows that when $k' = 60$ ELM-LRF has the best result (82.50%); and when $k_2 = 48$,

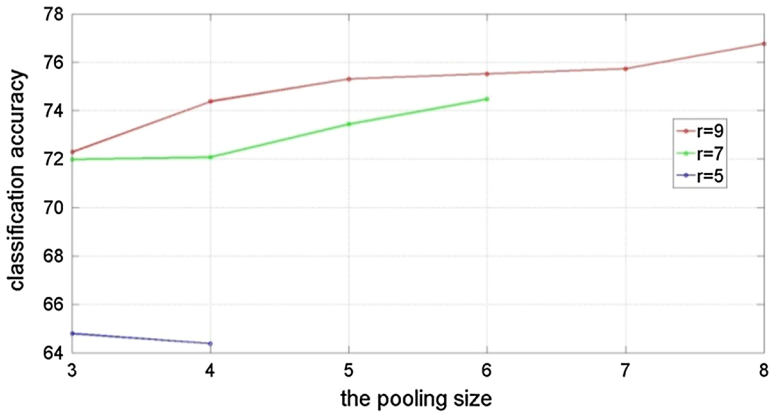


Fig. 9 Classification accuracy with different pooling size e, convolution size r of ELM-LRF when the number of the convolution filters k_2 is set to 48

Table 2 The classification accuracy (%) when ELM-HLRF and ELM-LRF are applied to dataset Outex_TC_00012 and k' is a constant

Methods	ELM-LRF ($k' = 48$)	ELM-HLRF ($k_2 = 48$)	
		$q = 4$	$q = 8$
Accuracy	76.77	90.63 ($p = 1$)	94.69 ($p = 1$)
		94.69 ($p = 2$)	96.56 ($p = 2$)
		95.31 ($p = 3$)	**96.88** ($p = 3$)
		94.90 ($p = 4$)	96.67 ($p = 4$)
		95.00 ($p = 5$)	96.77 ($p = 5$)

Bold value represents the best result of all the compared methods

$p = 3$ and $q = 4$, ELM-HLRF has the highest accuracy (96.77%). It can be seen that ELM-HLRF outperforms other methods in classification accuracy.

Figure 10 shows the classification results of ELM-LRF ($k' = 60$) and ELM-HLRF ($k_2 = 48$, $p = 5$ and $q = 4$) with different number of training samples. The classification accuracy of ELM-HLRF is always higher than that of ELM-LRF. Figure 11 gives the time consumption of ELM-LRF ($k' = 60$) and ELM-HLRF ($k_2 = 48$, $p = 5$ and $q = 4$) with varying number of training samples. It should be noted that the time here includes the time to produce convolution weights and the time to perform Gabor filtering procedure of ELM-HLRF in the training step. The time consumption of ELM-HLRF is more than ELM-LRF but ELM-HLRF has more convolution nodes and higher classification accuracy.

ELM-HLRF is also compared with the method presented by Yang et al. (2018). The average classification accuracy of the method (Yang et al. 2018) on Outex_TC_ 00012 is 96.54%, while ELM-HLRF has higher accuracy (96.88%).

4.3.2 Performance evaluation on dataset Outex_TC_00000

The classification results when ELM-LRF and ELM-HLRF are applied to dataset Outex_TC_00000 are shown in Tables 4 and 5. Several conclusions can be obtained from the two tables: first, Gabor filters are efficient convolution kernels when doing image feature

Table 3 The classification accuracy (%) when ELM-HLRF ($k_2 = 48$) and ELM-LRF are applied to dataset Outex_TC_00012 and $k' = (k_2 + p \cdot q)$

Methods	ELM-LRF		Gabor + ELM		Gabor + SVM		ELM-HLRF	
			$q = 4$	$q = 8$	$q = 4$	$q = 8$	$q = 4$	$q = 8$
Accuracy	70.10	75.00	27.92	33.75	36.25	44.48	90.63	94.69
	($k' = 52$)	($k' = 56$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)
	75.00	76.46	32.40	32.81	33.13	37.81	94.69	96.56
	($k' = 56$)	($k' = 64$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)
	82.50	74.38	28.33	32.08	34.58	35.42	95.31	**96.88**
	($k' = 60$)	($k' = 72$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)
	76.46	80.31	27.19	30.00	34.69	33.96	94.90	96.67
	($k' = 64$)	($k' = 80$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)
	81.15	81.04	27.60	29.17	30.52	32.92	95.00	96.77
	($k' = 68$)	($k' = 88$)	($p = 5$)	($p = 5$)	($p = 5$)	$p = 5$)	($p = 5$)	($p = 5$)

Bold value represents the best result of all the compared methods

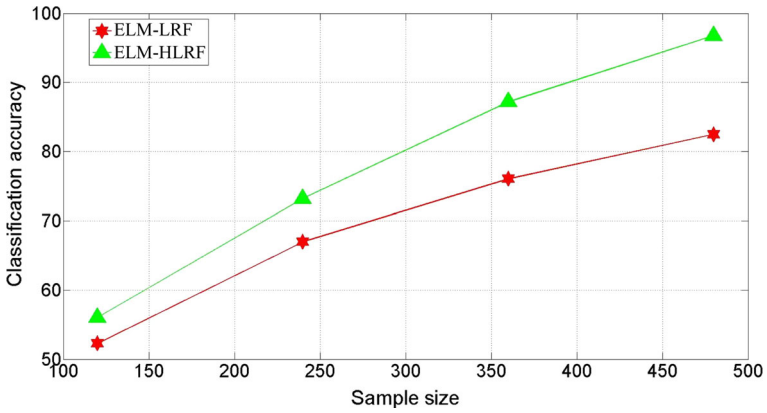


Fig. 10 Classification accuracy (%) with varying number of training samples when ELM-HLRF and ELM-LRF are applied to Outex_TC_00012

extraction; second, Table 5 shows that when $k' = (k_2 + p \cdot q)$ the classification accuracy of ELM-HLRF is higher than that of ELM-LRF, so the hybrid filter kernels are superior to the randomly generated convolution kernels. ELM-HLRF achieves its optimal performance when $k_2 = 48$, $p = 3$, $q = 8$ and does not increase with the increasing of p and q after then.

Figure 12 shows the classification results of ELM-LRF ($k' = 88$) and ELM-HLRF ($k_2 = 48$, $p = 5$ and $q = 8$) with different sizes of training samples. The classification accuracy of ELM-HLRF is always higher than that of ELM-LRF. Figure 13 gives the time consumption of ELM-LRF ($k' = 88$) and ELM-HLRF ($k_2 = 48$, $p = 5$ and $q = 8$) with varying number of training samples. The convolution nodes of ELM-LRF and ELM-HLRF are the same in this experiment, and it can be seen from Fig. 13 that the consumed time of ELM-LRF is more than that of ELM-HLRF.

The results of ELM-HLRF on Outex_TC_00000 are also compared with those of other methods. With the input images of the same size (32×32), ELM-HLRF has higher classifi-

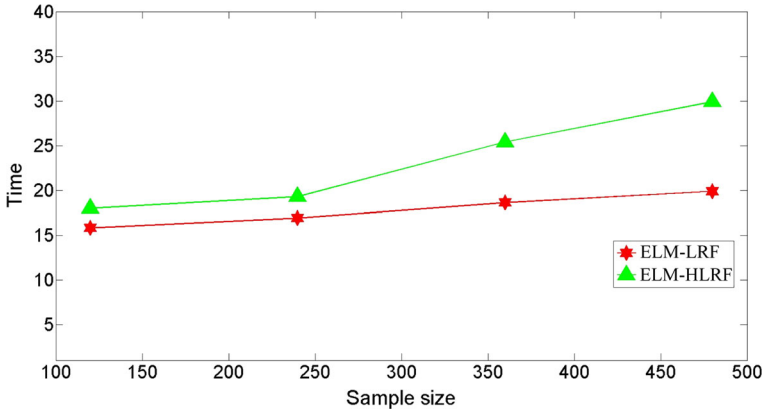


Fig. 11 Time-consumption (s) with varying number of training samples when ELM-HLRF and ELM-LRF are applied to Outex_TC_00012

Table 4 The classification accuracy (%) when ELM-HLRF and ELM-LRF are applied to dataset Outex_TC_00000 and $k' = 48$

Methods	ELM-LRF ($k' = 48$)	ELM-HLRF ($k_2 = 48$)	
		$q = 4$	$q = 8$
Accuracy	72.71	76.65 ($p = 1$)	78.78 ($p = 1$)
		79.98 ($p = 2$)	81.82 ($p = 2$)
		80.77 ($p = 3$)	**83.54** ($p = 3$)
		81.18 ($p = 4$)	83.51 ($p = 4$)
		81.07 ($p = 5$)	**83.54** ($p = 5$)

Bold values represent the best result of all the compared methods

Table 5 The classification accuracy (%) when ELM-HLRF ($k_2 = 48$) and ELM-LRF are applied to dataset Outex_TC_00000 and $k' = (k_2 + p \cdot q)$

Methods	ELM-LRF		Gabor + ELM		Gabor + SVM		ELM-HLRF	
			$q = 4$	$q = 8$	$q = 4$	$q = 8$	$q = 4$	$q = 8$
Accuracy	71.76	74.32	26.36	34.13	32.27	51.02	76.65	78.78
	($k' = 52$)	($k' = 56$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)	($p = 1$)
	74.32	76.88	31.66	39.29	49.77	59.33	79.98	81.82
	($k' = 56$)	($k' = 64$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)	($p = 2$)
	76.49	76.20	34.19	41.33	53.71	59.58	80.77	**83.54**
	($k' = 60$)	($k' = 72$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)	($p = 3$)
	76.88	79.33	35.96	41.76	53.19	57.18	81.18	83.51
	($k' = 64$)	($k' = 80$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)	($p = 4$)
	76.34	78.44	37.07	39.06	51.63	56.39	81.07	**83.54**
	($k' = 68$)	($k' = 88$)	($p = 5$)	($p = 5$)	($p = 5$)	($p = 5$)	($p = 5$)	($p = 5$)

Bold values represent the best result of all the compared methods

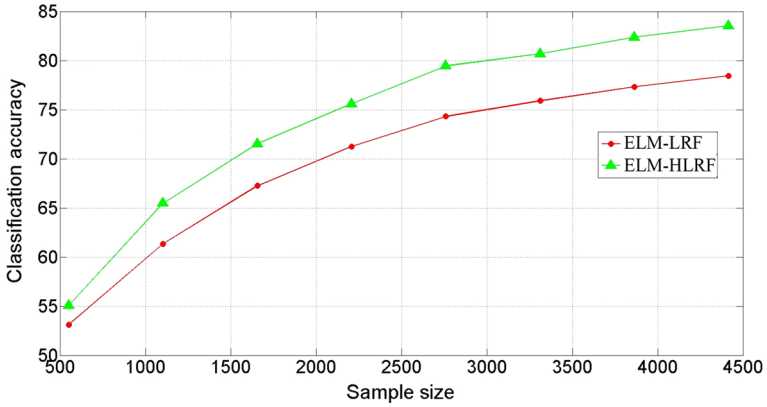


Fig. 12 Classification accuracy (%) with varying number of training samples when the proposed method and ELM-LRF are applied to Outex_TC_00000

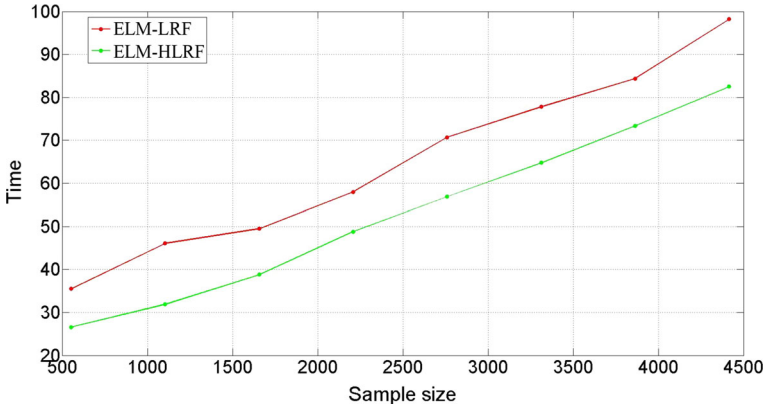


Fig. 13 Time-consumption (seconds) with varying number of training samples when ELM-HLRF and ELM-LRF are applied to Outex_TC_00000

cation accuracy (83.54%) than the method ((76.7 ± 1.8)%) presented by Reininghaus et al. (2015).

4.3.3 Performance evaluation on the Yale face database

Figure 14 shows the classification results when ELM-LRF is applied to the Yale face database. The number of training samples for each class is 5. We can see from Fig. 14 that the increasing number of convolution maps contributes little to the classification accuracy. So we set $k' = 4$ and $k' = 8$ in this paper for the consideration of time-consumption.

Table 6 shows the classification accuracy of ELM, SVM, ELM-LRF, the method presented by Zhang et al. (2014) and ELM-HLRF. When the number of convolution filters is set to 8 ($k_2 = 4$, $p = 1$ and $q = 4$), the classification accuracy of ELM-HLRF is higher than that of other methods. Specifically, when the number of convolution nodes of ELM-LRF and ELM-HLRF is the same, ELM-HLRF outperforms ELM-LRF.

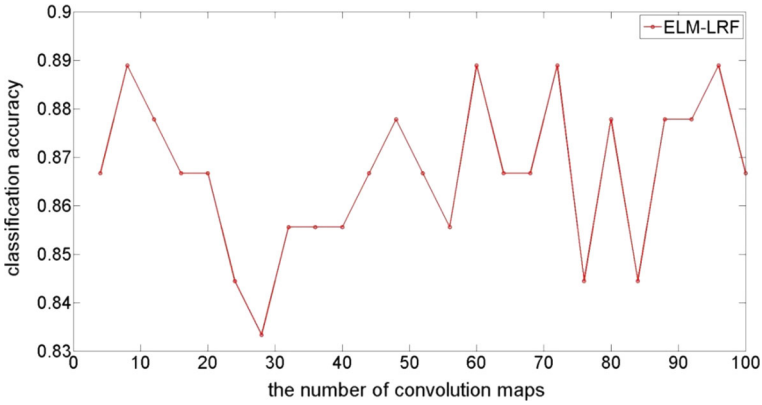


Fig. 14 The classification accuracy when ELM-LRF is applied to the Yale face database and the number of training maps for each class is 5

Table 6 The comparison of classification accuracy (%) on the Yale face database

Methods	The number of training images			
	5	6	7	8
ELM-LRF ($k' = 4$)	86.67	81.33	96.67	95.56
ELM-LRF ($k' = 8$)	88.89	84.00	96.67	**97.78**
Gabor + ELM ($p = 1, q = 4$)	15.56	14.67	16.67	22.22
Gabor + SVM ($p = 1, q = 4$)	28.89	25.33	28.33	20.00
Zhang et al. (2014)	89.40	92.35	93.87	95.51
ELM-HLRF ($k_2 = 4$, $p = 1, q = 4$)	**93.33**	**93.33**	**98.33**	**97.78**

Bold values represent the best result of all the compared methods

Table 7 The comparison of classification accuracy (%) and training time (s) on the ORL face database

Methods	Accuracy	Training time
Xu et al. (2015)	91.50	–
ELM-LRF ($k' = 24$, $r = 9$, $e = 8$, $C = 0.01$)	**98.50**	0.0548
ELM-HLRF ($k_2 = 20$, $p = 3$, $q = 4$, $r = 9$, $e = 8$, $C = 0.01$)	**98.50**	0.0711

Bold values represent the best result of all the compared methods

4.3.4 Performance evaluation on the ORL face database

The classification results of ELM-HLRF on the ORL face database are shown in Table 7. In this experiment, 5 images for each class are used for training and the rest 5 for testing. We compare ELM-HLRF with ELM-LRF and the image classification method presented by Xu et al. (2015). The optimal parameters of ELM-LRF and ELM-HLRF are shown in Table 7. ELM-HLRF and ELM-LRF have the same classification accuracy (98.50%), but the training time of ELM-HLRF is higher than that of ELM-LRF because the former has more convolution kernels.

Table 8 The parameters of ELM-LRF and ELM-HLRF on NORB

Methods	Parameters					
ELM-LRF	k'	r	e	C		
	48	4	3	0.01		
ELM-HLRF	k_2	p	q	r	e	C
	4, 8, …, 80	1, 2 ,3, 4, 5	4, 8	4	3	0.01

Table 9 The comparison of classification accuracy (%) and training time (s) on NORB

Methods	Accuracy	Training time
Random weights (Saxe et al. 2011)	95.20	1778.74
K-means + soft activation (Coates et al. 2011)	97.20	6977.18
Tiled CNN (Ngiam et al. 2010)	96.10	15522.83
CNN (LeCun et al. 2004b)	93.40	53815.57
DBN (Nair and Hinton 2009)	93.50	86419.56
ELM-LRF (Huang et al. 2015)	97.26	397.39
ELM-MSLRF (Huang et al. 2017)	**97.50**	403.95
ELM-HLRF ($k_2 = 48$, $p = 2$, $q = 8$)	**97.45**	516.08

Bold values represent the best result of all the compared methods

4.3.5 Performance evaluation on the NORB dataset

Table 9 shows the comparison of classification accuracy on the NORB dataset. The optimal parameters of ELM-LRF are shown in Table 8. The parameters of ELM-HLRF are shown in Table 8, and the optimal parameters are shown in Table 9. We can see from Table 9 that ELM-HLRF achieves good performance (97.45%), which is comparable to ELM-MSLRF (97.50%) and better than the other methods. Besides, the training time of ELM-HLRF is longer than that of ELM-LRF and ELM-MSLRF because the former has more convolution kernels.

4.3.6 Data augmentation

In this paper, data augmentation is considered to improve the overall classification accuracy. We use Gaussian blur to preprocess the training images first. Then the Gaussian blurred images combined with original training images will be provided as new training image inputs of ELM-HLRF or ELM-LRF. We use a 5×5 Gaussian blur function with standard deviation of 1 to preprocess the original training images. Table 10 shows the classification results when data augmentation is applied to the five datasets. It can be concluded that the results obtained by the proposed data augmentation method are better than those without data augmentation.

4.4 Discussions

Figures 15 and 16 show the relationship between the number of Gabor filters and the number of the random convolution kernels in ELM-HLRF on the Outex_TC_00012 dataset and the ORL face database respectively. We can see from Figs. 15 and 16 that, the accuracy increases

Table 10 Classification accuracy (%) of the three datasets before and after data augmentation

	Methods	Original	Data augmentation
Outex_TC_00012	ELM-LRF ($k' = 60$)	82.50	87.60
	ELM-HLRF ($k_2 = 48$, $p = 3$ and $q = 8$)	96.88	**97.29**
Outex_TC_00000	ELM-LRF ($k' = 88$)	78.44	80.22
	ELM-HLRF ($k_2 = 48$, $p = 5$ and $q = 8$)	83.54	**84.50**
Yale face dataset	ELM-LRF ($k' = 8$)	88.89	88.89
	ELM-HLRF ($k_2 = 4$, $p = 1$ and $q = 4$)	93.33	**95.56**
ORL face dataset	ELM-LRF ($k' = 24$)	98.50	**99.00**
	ELM-HLRF ($k_2 = 20$, $p = 3$ and $q = 4$)	98.50	**99.00**
NORB	ELM-LRF ($k' = 48$)	97.26	**97.33**
	ELM-HLRF ($k_2 = 48$, $p = 2$ and $q = 8$)	97.45	**97.52**

Bold values represent the best result of all the compared methods

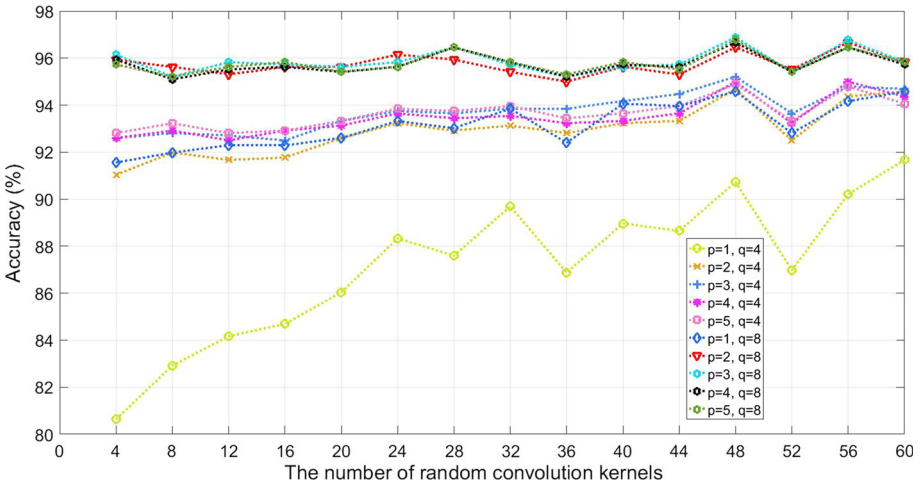


Fig. 15 The relationship between the number of Gabor filters and the number of random convolution kernels in ELM-HLRF on the Outex_TC_00012

with the increasing number of random convolution kernels at first, and then the number of random convolution kernels is not in evidence with the classification accuracy when $k_2 \geq 24$ for Outex_TC_00012 and $k_2 \geq 20$ for the ORL face database.

Furthermore, the results on the Outex_TC_00012, the ORL face database and the NORB dataset show that ELM-HLRF needs the more number of convolution kernels than ELM-LRF to achieve its highest classification accuracy. Therefore, ELM-HLRF spends more training time to achieve its optimal performance. On the other hand, ELM-HLRF has higher accuracy than ELM-LRF, which means that the Gabor filters can provide the features that the random convolution kernels cannot extract from images.

5 Conclusions

In this paper, we propose an innovative method local receptive field based extreme learning machine with hybrid filter kernels (ELM-HLRF) to carry out image classification. Two kinds

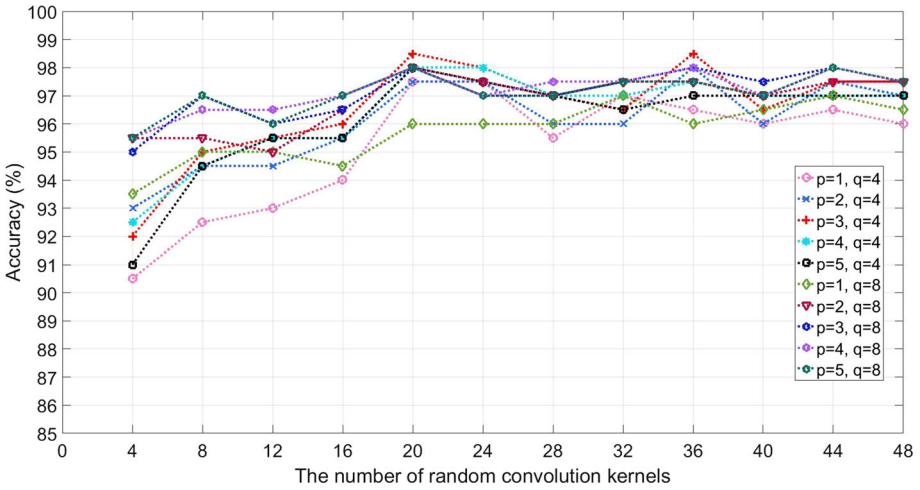


Fig. 16 The relationship between the number of Gabor filters and the number of random convolution kernels in ELM-HLRF on the ORL face database

of convolution kernels are included in ELM-HLRF: the kernels of ELM-LRF and Gabor filter kernels. In parallel, a data augmentation method based on Gaussian blur is used to improve classification performance. We evaluate the performance of ELM-HLRF and the data augmentation method using five datasets: Outex_TC_00000, Outex_TC_00012, the Yale face database, the ORL face database and the NORB dataset. It can be concluded that ELM-HLRF has higher classification accuracy than ELM-LRF, SVM and ELM, which proves that Gabor filter kernels are efficient convolution kernels. Also, the experiment results indicate that the training data augmented using data augmentation is more effective than the original training data.

Acknowledgements This work has been supported by The National Key Research and Development Program of China (2016YFC0301400) and Natural Science Foundation of China (51379198).

References

Bencherif, M. A., Bazi, Y., Guessoum, A., Alajlan, N., Melgani, F., & AlHichri, H. (2015). Fusion of extreme learning machine and graph-based optimization methods for active classification of remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, *12*(3), 527–531.

Chang, S.-Y. & Morgan, N. (2014). Robust CNN-based speech recognition with Gabor filter kernels. In *Fifteenth annual conference of the international speech communication association.*

Coates, A., Ng, A. & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cui, X., Goel, V., & Kingsbury, B. (2015). Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(9), 1469–1477.

Fogel, I., & Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, *61*(2), 103–113.

Georghiades, A., Belhumeur, P. & Kriegman, D. (1997). Yale face database. *Center for Computational Vision and Control at Yale University,2*, 6. http://cvc.yale.edu/projects/yalefaces/yalefa.

Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *6*, 610–621.

Huang, G.-B., Bai, Z., Kasun, L. L. C., & Vong, C. M. (2015). Local receptive fields based extreme learning machine. *IEEE Computational Intelligence Magazine*, *10*(2), 18–29.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics)*, *42*(2), 513–529.

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

Huang, J., Yu, Z. L., Cai, Z., Gu, Z., Cai, Z., Gao, W., et al. (2017). Extreme learning machine with multi-scale local receptive fields for texture classification. *Multidimensional Systems and Signal Processing*, *28*(3), 995–1011.

Jain, A. K., & Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, *24*(12), 1167–1186.

Jain, A. K., Ratha, N. K., & Lakshmanan, S. (1997). Object detection using gabor filters. *Pattern Recognition*, *30*(2), 295–309.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

LeCun, Y., Huang, F. J. & Bottou, L. (2004a). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004*, Vol. 2, IEEE, pp. II–104.

LeCun, Y., Huang, F. J. & Bottou, L. (2004b). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004*, Vol. 2, IEEE, pp. II–104.

LeCun, Y., Huang, F. J. & Bottou, L. (2004c). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004*, Vol. 2, IEEE, pp. II–104.

Li, W., Chen, C., Su, H., & Du, Q. (2015). Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(7), 3681–3693.

Manjunath, B. S., & Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 837–842.

Mehrotra, R., Namuduri, K. R., & Ranganathan, N. (1992). Gabor filter-based edge detection. *Pattern Recognition*, *25*(12), 1479–1494.

Nair, V. & Hinton, G. E. (2009). 3d object recognition with deep belief nets. In *Advances in neural information processing systems*, pp. 1339–1347.

Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V. & Ng, A. Y. (2010). Tiled convolutional neural networks. In *Advances in neural information processing systems*, pp. 1279–1287.

Ojala, T., Maenpaa, T., Pietikainen, M., Viertola, J., Kyllonen, J. & Huovinen, S. (2002). Outex-new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of 16th international conference on pattern recognition*, 2002, Vol. 1, IEEE, pp. 701–706.

Pietikäinen, M. (2010). Local binary patterns. *Scholarpedia*, *5*(3), 9775.

Rajadell, O., García-Sevilla, P., & Pla, F. (2013). Spectral-spatial pixel characterization using gabor filters for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, *10*(4), 860–864.

Reininghaus, J., Huber, S., Bauer, U. & Kwitt, R. (2015). A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4741–4748.

Samaria, F. S. & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of the second IEEE workshop on applications of computer vision*, 1994, IEEE, pp. 138–142.

Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B. & Ng, A. Y. (2011). On random weights and unsupervised feature learning. In *ICML*, pp. 1089–1096.

Xu, Y., Zhang, B., & Zhong, Z. (2015). Multiple representations and sparse representation for image classification. *Pattern Recognition Letters*, *68*, 9–14.

Yang, P., Zhang, F. & Yang, G. (2018). Fusing DTCWT and LBP based features for rotation, illumination and scale invariant texture classification. In *IEEE access*.

Zeng, Y., Xu, X., Shen, D., Fang, Y., & Xiao, Z. (2017). Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Transactions on Intelligent Transportation Systems*, *18*(6), 1647–1653.

Zhang, S., He, B., Nian, R., Wang, J., Han, B., Lendasse, A., et al. (2014). Fast image recognition based on independent component analysis and extreme learning machine. *Cognitive Computation*, *6*(3), 405–422.