



# Extreme learning machine with autoencoding receptive fields for image classification

Chao Wu<sup>1</sup> · Yaqian Li<sup>2</sup> · Zhibiao Zhao<sup>1</sup> · Bin Liu<sup>1</sup>

Received: 9 November 2018 / Accepted: 17 June 2019 / Published online: 26 June 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

Based on the theory of local receptive field based extreme learning machine (ELM-LRF) and ELM auto encoder (ELM-AE), a new network structure is proposed to take advantage of global attributes of image and output feature of each layer in the structure. This proposed network structure is called extreme learning machine with autoencoding receptive fields (ELM-ARF), which has two parts including convolution feature extraction and feature coding. In the convolution feature extraction part, local features are extracted using orthogonalized local receptive fields. The ELM-AE theory and local receptive fields are used to encode the global receptive fields, which is used to extract global features. The pooled global features and local features are combined and input into the next layer. In the feature coding part, the shallow layer feature can be input to any deep layer through the ELM-ARF connection structure. A series of encodings are performed on the combined features in each layer to achieve a nonlinear mapping relationship from input information to target categories. In order to verify the validity of the structure, ELM-ARF is tested on four classic databases: USPS, MNIST, NORB and CIFAR10. The experimental results show that ELM-ARF effectively improves image classification accuracy by encoding the combined features that contain global attributes.

**Keywords** Local receptive field based extreme learning machine (ELM-LRF) · ELM auto encoder (ELM-AE) · Local receptive fields · Global receptive fields · Image classification

## 1 Introduction

Currently, efficient machine learning algorithms [1, 2] are the focus of many researchers. To complete the training of the network quickly and efficiently without iteration, Huang et al. [3, 4] proposed a simple and effective extreme learning machine (ELM). In ELM, the connection weights as well as biases between the input layer and the hidden layer are randomly initialized, and the regularized least squares method is used to analytically determine the weight of the output layer. Huang et al. [5] extended ELM by introducing kernel learning and proposed an extreme

learning machine with kernel (KELM), which achieved better results by introducing the kernel function into ELM. In KELM, the selection step of the number of nodes is omitted by using the feature kernel mapping.

In some classification methods, feature extraction and selection play an important role before performing classification using classifiers [6, 7]. Both ELM and KELM are single-hidden-layer network structures, which could not handle image classification problems well [8]. Some improved algorithms based on ELM [8–10] have deepened the number of layers or changed the structure. These algorithms only improve their general approximation ability and do not improve the ability to extract important feature in the image. To solve the problems, Huang et al. [11] proposed a local receptive field based extreme learning machine (ELM-LRF), which is enable to process images directly by combining the concept of local receptive field with ELM. The input layer and hidden convolutional layer in ELM-LRF are locally connected, which allow the network to process the local structure of the image like convolutional neural networks (CNNs) [12–15]. Since the ELM-LRF does not

✉ Yaqian Li  
yaqianli@ysu.edu.cn

<sup>1</sup> Institute of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, Hebei, China

<sup>2</sup> Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, Hebei, China

require back-propagation (BP) algorithm [16] during training, the computation and time required for training are greatly reduced.

ELM-LRF is improved in two ways. On one hand, ELM-LRF has only one convolutional layer and one pooling layer, and its performance is limited by the shallow architecture. Therefore, researchers propose to improve it by using multiple sets of convolutional and pooling layers to obtain more abstract feature representations, such as CKELM [17] and DC-ELM [18], and they prove that multiple set of convolutional and pooling layers could effectively improve classification accuracy. However, DC-ELM simply increases the number of convolutional layers and does not consider the identifiable category details contained in the shallow features, resulting in a waste of feature. On the other hand, single-scale local receptive fields are used in ELM-LRF and it is not suitable for complex texture image classification. Liu et al. [19] proposed a multi-mode ELM-LRF (MM-LRF-ELM) framework. In MM-LRF-ELM, the feature in each RGB channel and depth feature are, respectively, extracted, and then, the above features are used to construct the nonlinear feature. Huang et al. [20] proposed an extreme learning machine with multi-scale local receptive fields (ELM-MSLRF) by combining several local receptive fields of different sizes. Compared with ELM-LRF, ELM-MSLRF obtains a better result in texture image classification. He et al. [21] propose an extreme learning machine with hybrid local receptive fields (ELM-HLRF), which uses the Gabor function as a convolutional kernel filter. Gabor filters with different scales and directions are used to extract more features in ELM-HLRF. Extracted features are combined with local receptive field features to improve classification accuracy. Due to the limitation of local receptive field size and the way of convolution, features with global attributes cannot be extracted by the above methods.

Because of the above two issues, this paper proposes a new network structure extreme learning machine with autoencoding receptive fields (ELM-ARF). ELM-ARF is mainly improved in two aspects. Firstly, the theory of ELM auto encoder (ELM-AE) [22] and local receptive fields is used to train global receptive fields, which are used to extract the object contour in the image. After being pooled separately, global receptive field features are concatenated with the local receptive field features and input into next layer. Then, by using the identity mapping theory in residual network (ResNet) [23], the shallow layers are directly connected to the deep layers, so that the category details contained in the shallow layer feature can flow to any deep layer in the network. By performing a series of combined coding on the feature extracted from each layer, ELM-ARF makes full use of the feature contained in the image to effectively improve the accuracy of image classification.

The main content of this paper is organized as follows. The theory of ELM-AE is introduced in Sect. 2. The structure and training processes of ELM-ARF are introduced in Sect. 3. Section 4 presents our experimental results on USPS, MNIST, NORB and CIFAR10 databases. The paper concludes in Sect. 5.

## 2 Related works

The main objective of ELM-AE is to make three different equivalent representations of the original input features: (1) compressed representation, the input feature is equivalently mapped from the original feature space to the low-dimensional feature space; (2) sparse representation, the input feature is equivalently mapped from the original feature space to the high-dimensional feature space; and (3) equal dimension representation, the input features are equivalently mapped from the original feature space to the equal-dimensional feature space.

The equivalent representation of ELM-AE can be implemented in two steps.

In the first step, the mapping matrix  $\beta$  is obtained by using the principle of ELM training output weight matrix.  $X \in \mathbb{R}^{N \times n_i}$  is the input sample matrix,  $N$  is the number of samples, and  $n_i$  is the dimension of samples. The input weight  $a \in \mathbb{R}^{n_i \times n_h}$  between the input layer and the hidden layer is randomly initialized, and the hidden-layer offset is  $b \in \mathbb{R}^{N \times n_h}$ . Then, the output feature matrix of the hidden layer is  $H = g(Xa + b) \in \mathbb{R}^{N \times n_h}$ , where  $n_h$  represents the dimension of the hidden-layer feature and  $g(\cdot)$  is the activation function.  $X$  is used as an output feature to train the network output weight  $\beta$ , and then  $X = H\beta$ .  $\beta$  can be calculated by:

$$\beta = \begin{cases} \left( \frac{I_{n_h}}{C} + H^T H \right)^{-1} H^T X & N > n_h \\ H^T \left( \frac{I_N}{C} + H H^T \right)^{-1} X & N < n_h \end{cases} \quad (1)$$

In the second step, the input sample features are equivalently mapped by using the trained mapping matrix  $\beta$  to obtain the equivalent features  $X_{\text{out}} = X\beta^T$ ,  $X_{\text{out}} \in \mathbb{R}^{N \times n_h}$ . When  $n_i > n_h$ , the compression equivalent representation is implemented; when  $n_i < n_h$ , the sparse equivalent representation is implemented; when  $n_i = n_h$ , the equal-dimensional equivalent representation is implemented.

## 3 Network structure of ELM-ARF

In order to extract feature with global attributes and to utilize the category details contained in the shallow layer feature, the ELM-ARF consists of a convolution feature extraction part and a feature coding part, as shown in Fig. 1.

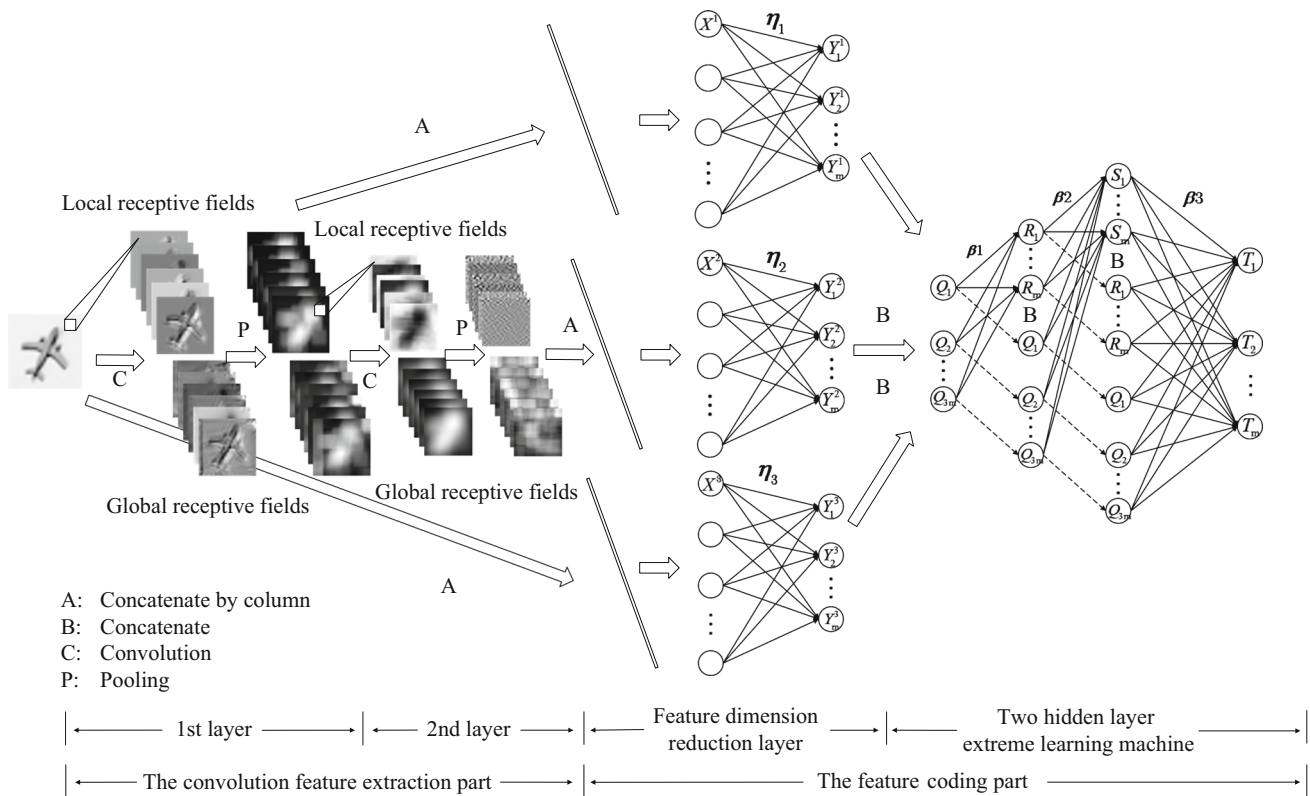


Fig. 1 Architecture of ELM-ARF

### 3.1 Convolution feature extraction

In the convolution feature extraction part, the global receptive fields are encoded by using the theory of ELM-AE. The convolution features are extracted from the local receptive fields and the global receptive fields. After being pooled separately, the above two features are concatenated to fuse different receptive field features, and these fused features are input into the next convolutional layer.

#### 3.1.1 Autoencoding of global receptive field

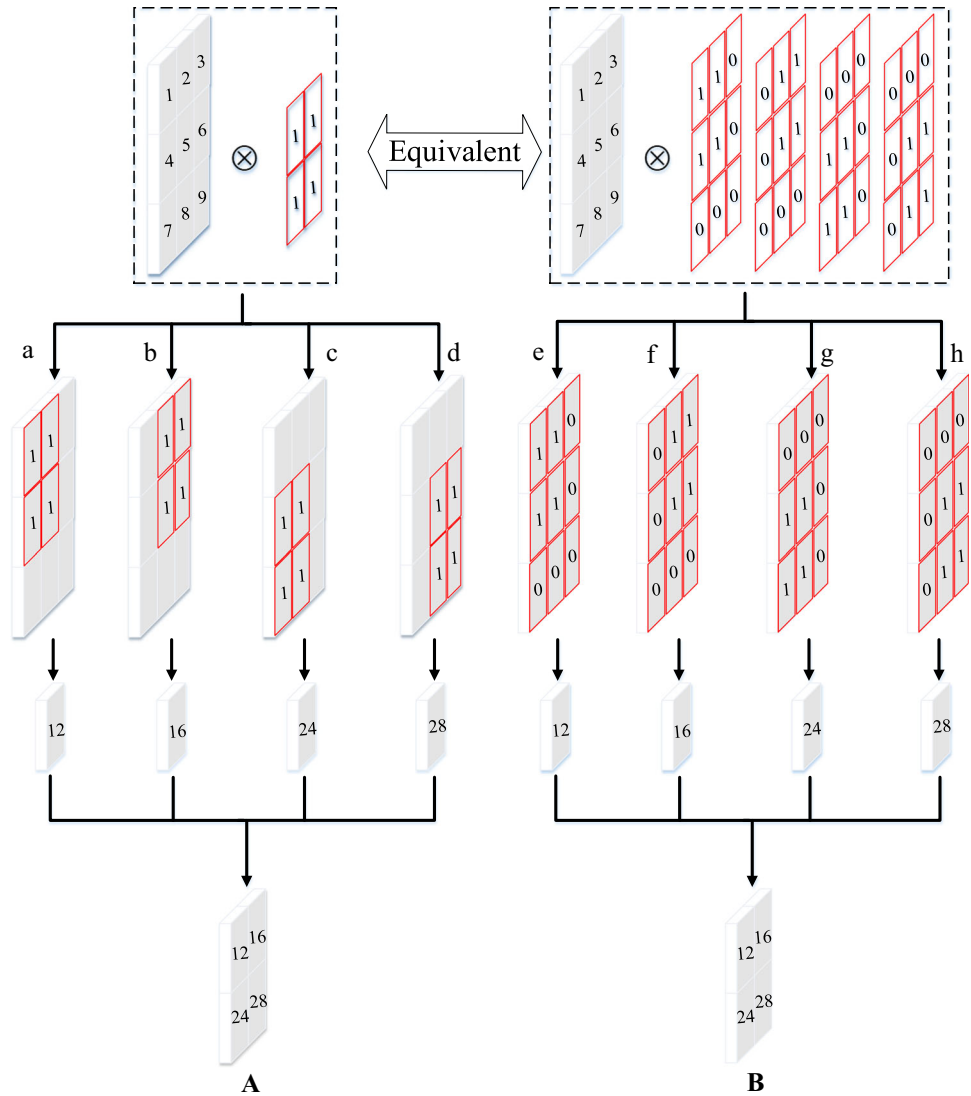
The global receptive field is trained by utilizing the theory of ELM-AE and local receptive fields. During ELM-AE training, a randomly initialized input weight matrix is required. Since the local receptive fields have obvious advantages in the extraction of image features [11], we use the local receptive fields that are also randomly initialized as the input weight to train the global receptive field. Because the local receptive fields are not in the form of a matrix, it cannot be used directly. Therefore, the local receptive fields need to be equivalently transformed into the form of a weight matrix for training. The equivalent transformation is shown in Figs. 2 and 3.

As shown in branch A of Fig. 2, a  $3 \times 3$  matrix is convoluted with a  $2 \times 2$  local receptive field. The

convolution step is shown in the middle part of the branch A, and the local receptive field generates a feature map with a size of  $2 \times 2$  in the sliding order of a, b, c and d. Branch B indicates that the  $3 \times 3$  matrix is convoluted with four  $3 \times 3$  receptive fields. In the receptive field, the matrix value at the position of the coefficient 0 has no effect on the generated convolution value. Therefore, each step of the convolution operation in the branch A can be equivalent to that of the corresponding  $3 \times 3$  receptive field in the branch B. The generated convolution values of branch B are arranged in a matrix according to the convolution order in branch A, and the matrix is the same as the matrix generated by the branch A. Therefore, the convolution operation of the branch A can be equivalently represented by the branch B.

In Fig. 3, each column of the  $3 \times 3$  matrix is concatenated to generate a column vector of  $9 \times 1$ , and the four receptive fields (e, f, g and h) are, respectively, concatenated to generate column vectors. These column vectors are transposed into row vectors and merged into a matrix with a convolution order of e, g, f and h. Then, the convolution operation of the branch B in Fig. 2 can be equivalent to the product operation of the two matrices in Fig. 3. Therefore, in this paper, the local receptive fields are extended to the weight matrix. The ELM-AE method is utilized to train the global receptive field matrix.

**Fig. 2** Convolution step. **A** represents convolution step of  $2 \times 2$  local receptive field, and the convolution order is a, b, c and d. **B** represents convolution step of  $3 \times 3$  receptive fields, and the convolution order is e, f, g and h

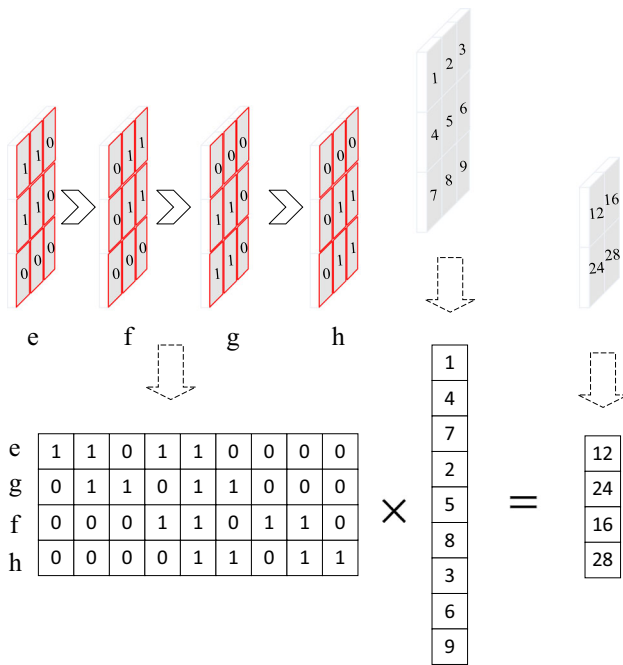


Suppose the size of the input image is  $d \times d$ , and the size of the local receptive field is  $r \times r$ , then the size of the output feature map is  $(d - r + 1) \times (d - r + 1)$ . The input weight matrix  $\hat{A}^{init} \in \mathbb{R}^{r^2 \times g}$  is randomly initialized, and  $g$  is the number of local receptive fields.  $\hat{A}^{init} \in \mathbb{R}^{r^2 \times g}$  is orthogonalized using singular value decomposition (SVD) method to generate a matrix  $\hat{A} \in \mathbb{R}^{r^2 \times g}$ . Let  $F = (d - r + 1)^2 \cdot g$ ,  $\hat{A}$  is extended to the weight matrix  $W \in \mathbb{R}^{d^2 \times F}$  according to the concept shown in Figs. 2 and 3. Then, the hidden-layer output matrix  $H = XW$ , and the global receptive field matrix  $\hat{B} \in \mathbb{R}^{d^2 \times F}$  is trained by using formula (1). The global receptive field matrix to the  $k$ -th feature map is  $\hat{B}_k \in \mathbb{R}^{d^2 \times (d-r+1)^2}, k = 1, 2, \dots, g$ . Each column in  $\hat{B}_k \in \mathbb{R}^{d^2 \times (d-r+1)^2}, k = 1, 2, \dots, g$  is changed into a receptive field form. These receptive fields, which are arranged in the convolution order of the branch A of

Fig. 2, to the  $k$ -th feature map are the  $k$ -th global receptive field  $B_k \in \mathbb{R}^{d \times d \times (d-r+1)^2}, k = 1, 2, \dots, g$ . Therefore, the number of global receptive fields is the same as the number of local receptive fields, and the size of the two types of feature maps is the same.

### 3.1.2 Convolution and pooling operation

In Fig. 1, the first layer and the second layer are, respectively, composed of a convolutional layer and a pooling layer. Local receptive fields and global receptive fields are used to extract features in two convolutional layers. Similar with ELM-LRF [11],  $A^L \in \mathbb{R}^{r^2 \times g}, L = 1, 2$  is used to equivalently represent the local receptive fields, where  $L$  is the layer number of convolutional layers,  $r$  is the local receptive fields size,  $g$  is the number of local receptive fields, and  $a_k^L \in \mathbb{R}^{r \times r}, k = 1, 2, \dots, g$  is the  $k$ -th local receptive field of the  $L$ -th layer.



**Fig. 3** Equivalent of the convolution step

The size of the input image  $X^{L-1}$  is  $d \times d$ , and the size of the output feature map is  $(d - r + 1) \times (d - r + 1)$ . The convolutional node  $(i, j)$  in the feature map of the  $k$ -th local receptive field is calculated as:

$$l_{i,j,k}^L(X) = \sum_{u=1}^r \sum_{v=1}^r x_{i+u-1,j+v-1} \cdot a_{u,v,k}, i, j = 1, \dots, (d - r + 1) \tag{2}$$

The global receptive fields  $B_k^L \in \mathbb{R}^{d \times d \times (d-r+1)^2}, k = 1, 2, \dots, g, L = 1, 2$  are trained according to Sect. 3.1.1.  $b_{i,j,k}^L \in \mathbb{R}^{d \times d}$  is the receptive field corresponding to the convolutional node  $(i, j)$  in the  $k$ -th feature map of the  $L$ -th layer. The convolutional node  $(i, j)$  in the feature map of the  $k$ -th global receptive field is calculated as:

$$g_{i,j,k}^L(X) = \sum_{u=1}^d \sum_{v=1}^d x_{u,v} \cdot b_{u,v,i,j,k}, i, j = 1, \dots, (d - r + 1) \tag{3}$$

Then, the generated  $l^L$  and  $g^L$  are, respectively, input into the pooling layer of size  $e$ , and the combinatorial node  $(p, q)$  in the  $k$ -th pooling map of local and global receptive field is, respectively, calculated as:

$$lh_{p,q,k}^L = \sqrt{\sum_{i=p-e}^{p+e} \sum_{j=q-e}^{q+e} (l_{i,j,k}^L)^2}, p, q = 1, \dots, (d - r + 1) \tag{4}$$

$$gh_{p,q,k}^L = \sqrt{\sum_{i=p-e}^{p+e} \sum_{j=q-e}^{q+e} (g_{i,j,k}^L)^2}, p, q = 1, \dots, (d - r + 1) \tag{5}$$

The generated  $lh^L$  and  $gh^L$  are concatenated into  $X^L = [lh^L, gh^L]$ .  $X^L$  is input to the next layer and repeats the above operations to fully fuse the local feature with the global feature.

**Algorithm 1** Convolution feature extraction

**Input:** Input image  $X^0$ , image size  $d$ , local receptive fields size  $r$ , number of receptive fields  $g$

**Output:** Convolution feature  $X^L (L=1,2)$ , local receptive fields  $A^L \in \mathbb{R}^{r \times r \times g}$ , global receptive fields  $B_k^L \in \mathbb{R}^{d \times d \times (d-r+1)^2}$

Step 1: for  $L = 1 : 2$  do:

Step 2: Randomly initialize the input weight matrix  $\hat{A}^{\text{init}} \in \mathbb{R}^{r^2 \times g}$

Step 3: Generate orthogonal matrix  $\hat{A}^L \in \mathbb{R}^{r^2 \times g}$  using SVD method

Step 4: Extend  $\hat{A}^L \in \mathbb{R}^{r^2 \times g}$  to the weight matrix  $W^L \in \mathbb{R}^{d^2 \times F}$ ,  $H^L = X^{L-1} W^L$

Step 5: Train matrix  $\hat{B}^L \in \mathbb{R}^{d^2 \times F}$  using formula (1)

Step 6: Extract local feature map  $l^L$  using local receptive field  $A^L \in \mathbb{R}^{r \times r \times g}$

Step 7: Extract global feature map  $g^L$  using global receptive field  $B_k^L \in \mathbb{R}^{d \times d \times (d-r+1)^2}$

Step 8: Calculate  $lh^L$  and  $gh^L$  using formula (4) and (5)

Step 9: Concatenate  $lh^L$  and  $gh^L$ , generate  $X^L = [lh^L, gh^L]$

Step 10: Update  $d = d - r + 1$

**Return**  $X^L (L=1,2)$ ,  $A^L \in \mathbb{R}^{r \times r \times g}$ ,  $B_k^L \in \mathbb{R}^{d \times d \times (d-r+1)^2}$

### 3.2 Feature coding

In the feature coding part, the dimensions of the convolution features of each layer are reduced, and then, the low-dimensional features are input into the final two-hidden-layer extreme learning machine for encoding and classification.

#### 3.2.1 Feature dimension reduction

In order to make full use of the identifiable category details in the shallow layer features, the input image, the first layer output feature maps and the second layer feature maps are, respectively, concatenated to generate a feature vector. Image or feature maps are concatenated by columns to generate high-dimensional feature vectors. For example, the image size of the NORB database is  $32 \times 32 \times 2$ , and the generated feature vector is 2048-dimension. In order to reduce the dimension while encoding, these three vectors are, respectively, multiplied by its corresponding matrices of the feature dimension reduction layer. And sample labels are used to train the weight matrix  $\eta_i (i = 1, 2, 3)$  according to the theory of ELM. Let  $X_1 = X^1, X_2 = X^2$  and  $X_3 = X^0$ , three weight matrices can be trained by formula (6), where  $i = 1, 2, 3, T \in \mathbb{R}^{N \times m}$  is the label matrix corresponding to the input image, and  $m$  is dimension of  $T$ .  $N$  is the number of features in  $X_i$ , and  $P_i$  is the dimension of the features in  $X_i$ .

$$\eta_i = \begin{cases} \left( \frac{I_{P_i} + X_i^T X_i}{C} \right)^{-1} X_i^T T & N > P_i \\ X_i^T \left( \frac{I_N + X_i X_i^T}{C} \right)^{-1} T & N < P_i \end{cases} \quad (6)$$

Then, the output of the feature dimension reduction layer is  $Y_i = X_i \eta (i = 1, 2, 3)$ , and  $Y_i \in \mathbb{R}^{N \times m}$ . The label vector of the NORB database is 5 dimensions. After the encoding of  $\eta_i (i = 1, 2, 3)$ , the dimension of output feature  $Y_i = X_i \eta (i = 1, 2, 3)$  is reduced to 5, which effectively reduces the calculation amount of the subsequent equivalent encoding.

#### 3.2.2 Two-hidden-layer extreme learning machine

After dimension reduction step, features  $Y_1, Y_2$  and  $Y_3$  are concatenated, and  $Q = [Y_1, Y_2, Y_3]$  is input into the two-hidden-layer ELM which is used to combine and encode features of each layer. Among them,  $\beta_1$  and  $\beta_2$  are used to encode features, and  $\beta_3$  is used to classify final features.

The dotted line in the two-hidden-layer ELM indicates that the input feature is concatenated with the output feature. For example,  $R = Q\beta_1$  and the input of the  $\beta_2$  layer is  $[R, Q]$ .  $S = [R, Q]\beta_1$ , and the input of the  $\beta_3$  layer is  $[S, R, Q]$ , where  $S \in \mathbb{R}^{N \times m}, R \in \mathbb{R}^{N \times m}, Q \in \mathbb{R}^{N \times 3m}$ . It can be observed that this connection structure can ensure that the features of each layer can flow to deeper layers in the structure, so that the features in the first layer can be utilized while the  $\beta_3$  layer utilizes the features  $[S, R, Q]$ .

In order to enable  $\beta_1$  and  $\beta_2$  to perform dimensionality reduction while encoding features,  $\beta_1$  and  $\beta_2$  are trained using the sample label  $T$ . Let  $H_1 = Q, H_2 = [R, Q], H_3 = [S, R, Q]$ .  $N$  is the number of features  $H_i$ , and  $M_i$  is the dimension of the features  $H_i$ .

$$\beta_i = \begin{cases} \left( \frac{I_{M_i} + H_i^T H_i}{C} \right)^{-1} H_i^T T & N > M_i \\ H_i^T \left( \frac{I_N + H_i H_i^T}{C} \right)^{-1} T & N < M_i \end{cases} \quad (7)$$

Better accuracy is obtained using kernel mapping in the classification layer of  $\beta_3$ . By transforming the matrix product in the formula into a kernel function, the mapping of features from low-dimensional space to high-dimensional space is realized. The traditional KELM only has a form of formula used when  $N < M_3$ . However, when  $N > M_3$ , the above formula will produce a high-dimensional square matrix. For example, MNIST has 60,000 sample features, and the sample dimension is only 784. Formula used when  $N < M_3$  will produce a square matrix of  $60000 \times 60000$ . The inversion of the high-dimensional square matrix will significantly increase the amount of calculation. In order to improve the classification accuracy and avoid the generation of high-dimensional matrix, when  $N > M_3$ , we replace the  $H_3^T H_3$  in  $\beta_3$  with the Gaussian radial basis kernel function (8) to realize the partial nuclear mapping function of KELM, as shown in formula (9).

$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{\sigma^2}\right), \sigma > 0 \quad (8)$$

$$\beta_3 = \left( \frac{I_{M_3} + K(H_3^T, H_3^T)}{C} \right)^{-1} H_3^T T \quad (9)$$

The input test feature of the  $\beta_3$  layer is  $h_3$ , and the output prediction value of the two-hidden-layer ELM is  $t = h_3 \beta_3$ . When  $N < M_3$ , we use the traditional form of KELM, and the predicted value is calculated as:

$$t = K(h_3, H_3) \left( \frac{I_N + K(H_3, H_3)}{C} \right)^{-1} T \quad (10)$$

**Algorithm 2** Feature coding

**Input:**  $X^L (L=0,1,2)$   
**Output:** Weight matrix  $\eta_i (i=1,2,3)$ ,  $\beta_i (i=1,2,3)$   
 Step 1: Let  $X_1 = X^1, X_2 = X^2$  and  $X_3 = X^0$   
 Step 2: Calculate  $\eta_i (i=1,2,3)$  using formula (6)  
 Step 3: Calculate  $Y_i = X_i \eta_i (i=1,2,3)$   
 Step 4: Let  $H_1 = Q = [Y_1, Y_2, Y_3]$ , calculate  $\beta_1$  using formula (7)  
 Step 5: Calculate  $R = H_1 \beta_1$ , let  $H_2 = [R, Q]$   
 Step 6: Calculate  $\beta_2$  using formula (7)  
 Step 7: Calculate  $S = H_2 \beta_2$ , let  $H_3 = [S, R, Q]$   
 Step 8: Calculate  $\beta_3$  using formula (9)  
**Return**  $\eta_i (i=1,2,3)$ ,  $\beta_i (i=1,2,3)$

**3.3 Time and space complexities**

The structure in Fig. 1 is used as an example to analyze the time and space complexities of the ELM-ARF. The training sample matrix is  $X \in \mathbb{R}^{N \times d^2}$ , and the test sample matrix is  $X_1 \in \mathbb{R}^{M \times d^2}$ . The local receptive field size is  $r$ . The number of both local receptive fields and global receptive fields per layer is  $g$ . The pooling size is  $e$ . The output feature matrix of the first layer is  $X^1 \in \mathbb{R}^{N \times l_1 g}$ , where  $l_1 = (d - r + 1)^2$ . The second layer output feature matrix is  $X^2 \in \mathbb{R}^{N \times l_2 g}$ , where  $l_2 = (d - 2r + 2)^2$ . The label matrix of the training sample is  $T \in \mathbb{R}^{N \times m}$ , and the label matrix of the testing sample is  $T_1 \in \mathbb{R}^{M \times m}$ . The time and space complexities of the single-layer ELM-LRF are used to compare with that of ELM-ARF. In the single-layer ELM-LRF, the number of local receptive fields is  $4g$ , and other parameters are the same as ELM-ARF.

In the training stage, the time complexity of ELM-ARF is  $O(N(6l_1^2g^2 + 6l_2^2g^2 + 8l_1l_2g^2 + 4d^2l_1g + 2d^4) + 9(l_1^3 + l_2^3)g^3 + d^6)$ .  $l_1$  and  $l_2$  are amplified to  $d^2$ . Assume  $d^2 = l_1 = l_2 = pN$ , where  $p \ll 1$ . The training time complexity of ELM-ARF can be approximated as  $O(((4g + 20g^2)p^2 + 18g^3p^3)N^3)$ . The training time complexity of the ELM-LRF can be approximated as  $O(((4g + 32g^2)p^2 + 64g^3p^3)N^3)$ . By adding a convolution feature extraction layer and a feature dimension reduction layer to reduce the amount of computation, the ELM-ARF has a lower training time complexity.

In the testing stage, the time complexity of ELM-ARF is  $O(M(2d^2l_1g + 4l_1l_2g^2))$ , which can be approximated as  $O((2 + 4g)gp^2M^3)$ . The testing time complexity of the ELM-LRF can be approximated as  $O(4gp^2M^3)$ . Compared to ELM-LRF, ELM-ARF has more structure, which contains more calculations during testing. Therefore, ELM-ARF has a higher testing time complexity.

The space complexity of ELM-ARF is  $O(N(3l_1g + 3l_2g + d^2) + 2d^2l_1g + 4l_1l_2g^2 + d^2l_2)$ , which can be approximated as  $O((8 + 4g)gp^2N^2)$ . The space complexity of the ELM-LRF can be approximated as  $O(8gp^2N^2)$ . Compared to ELM-LRF, ELM-ARF needs to store more weight matrices. Therefore, it has a higher space complexity.

**4 Experiments**

In order to verify the validity of ELM-ARF, we carry out experiments in USPS [24], MNIST [25], NORB [26] and CIFAR10 [27] databases, and the experimental results are compared with the results of some convolutional networks trained based on the ELM method. The experimental environment is the supercomputing system in the High Performance Computing Center of Yanshan University, whose specific hardware is 1 Intel E5-2683v3 CPU (28 cores 2.0 Ghz), 64 GB memory per node. We use resource scheduling instructions to occupy 1 node (28 cores, 64 GB). The operating system and software environment are Centos7.2, MATLAB R2018a.

**4.1 USPS database**

USPS is a handwritten digital recognition database containing a total of 9298 images, which contain ten numbers from 0 to 9. Example images in the database are shown in Fig. 4. The training sample image is 7291, and the test sample image is 2007. The number in the image is centered, and the images are all normalized to  $16 \times 16$  pixels. The database has small number of samples and is relatively simple, so it is first used to verify the validity of ELM-ARF. We select all training samples and test samples for experimentation.

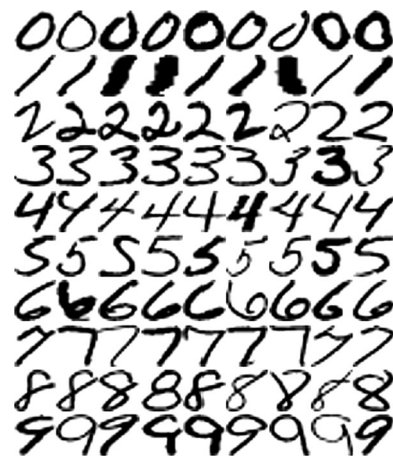
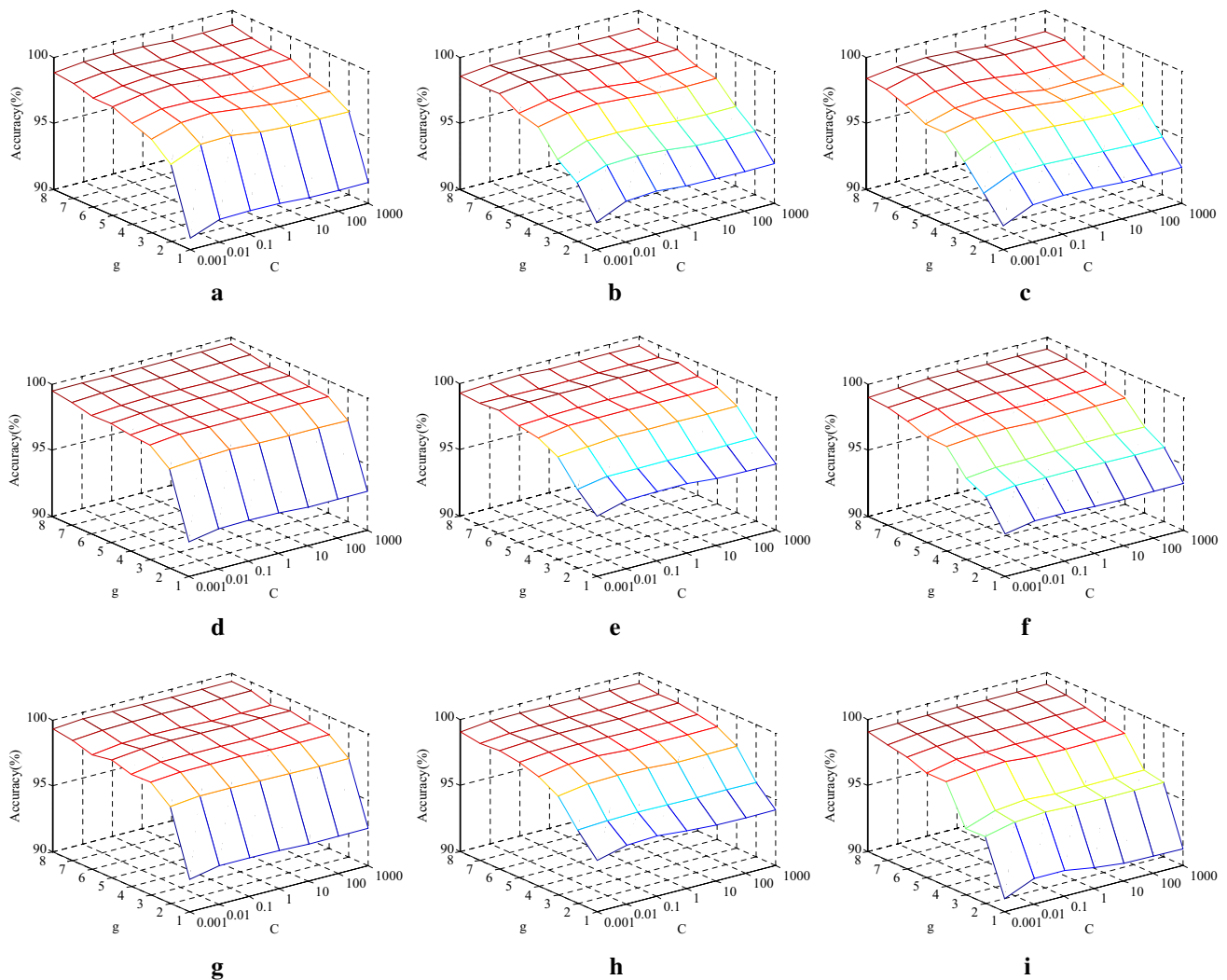


Fig. 4 Example images in the USPS database



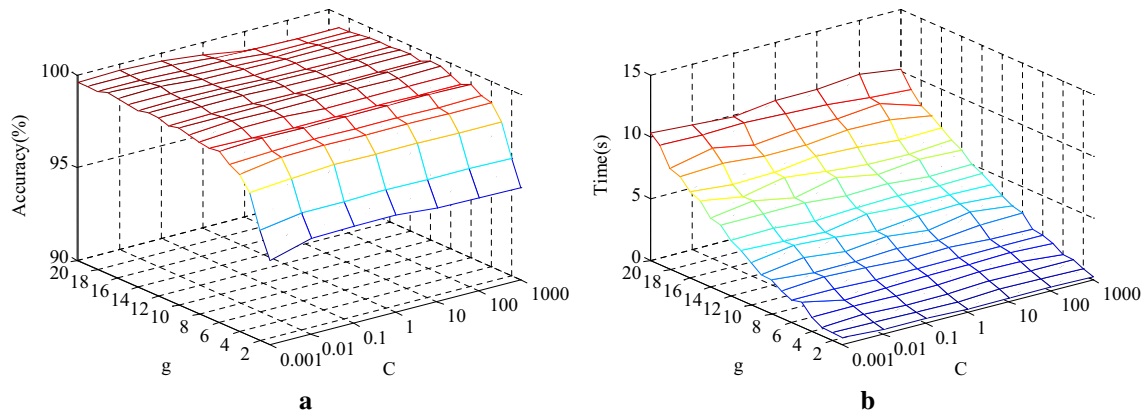
**Fig. 5** ELM-ARF parameters selection on USPS database. **a** 1 layer and  $3 \times 3$  LRF, **b** 1 layer and  $4 \times 4$  LRF, **c** 1 layer and  $5 \times 5$ , **d** 2 layers and  $3 \times 3$ , **e** 2 layers and  $4 \times 4$ , **f** 2 layers and  $5 \times 5$ , **g** 3 layers and  $3 \times 3$ , **h** 3 layers and  $4 \times 4$ , **i** 3 layers and  $5 \times 5$

For the USPS database, we need to select the optimal network parameters for the ELM-ARF. These parameters include the size of local receptive fields, the number of layers, the number of receptive fields per convolutional layer and the penalty coefficient. The size of local receptive fields is, respectively, set to  $3 \times 3$ ,  $4 \times 4$ ,  $5 \times 5$ . The number of layers  $i$  is set to  $\{1, 2, 3\}$ . The number of local receptive fields is equal to the number of global receptive fields, and the number of both per convolutional layer is represented by  $g$ . The parameter  $g$  is set to  $\{1, 2, \dots, 8\}$ . The number of receptive fields per convolutional layer is  $2 \times g$ , and the total number of receptive fields is  $2 \times g \times i$ . The penalty coefficient  $C$  is set to  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . The pooling layer size is  $3 \times 3$ , which is consistent with the literature [11].

The accuracy of ELM-ARF is changed as the parameters change. Figure 5 shows the accuracy mesh of ELM-ARF with different numbers of layers and different sizes of local

receptive fields. The mesh diagrams with the same number of layers are placed on the same row, and the number of layers increases from top to bottom. The mesh diagrams with the same local receptive field size are placed in the same column, from left to right, and the size is  $3 \times 3$ ,  $4 \times 4$  and  $5 \times 5$ , respectively. All mesh diagrams show that the accuracy is changed with the change of  $g$  and  $C$ . It can be observed that the classification accuracy of the network decreases as the size of local receptive fields increases. When the number of layers is 2 and local receptive fields size is  $3 \times 3$ , the ELM-ARF can achieve the highest accuracy in Fig. 5. When the number of layers is 2 and the size of local receptive fields is  $4 \times 4$ , the average accuracy is the highest and the mesh is smoother. The USPS database is relatively simple. When the network is added to 2 layers, the accuracy has reached 99.5%. The accuracy increase is not obvious when the network is added





**Fig. 6** Accuracy and time of ELM-ARF (2 layers,  $4 \times 4$  and  $3 \times 3$ ) on USPS database

**Table 1** Comparison of accuracies and time on USPS database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
ELM-LRF	2.16	0.44	98.54
CKELM	1.4	0.4	98.46
DC-ELM	2.4	0.8	98.46
ELM-MSLRF	2	0.6	98.74
ELM-ARF	1.7	0.47	<b>99.49</b>

to three layers, but the training time is significantly increased. Therefore, the number of layers of ELM-ARF is set to 2.

ELM-ARF can obtain better experimental results when the local receptive field size is  $3 \times 3$  or  $4 \times 4$ . Therefore, these two sizes are used in combination. The first layer uses local receptive fields of size  $4 \times 4$ , and the second layer uses local receptive fields of  $3 \times 3$ . In order to test the effect of this combination, the parameter  $g$  is set to  $\{1, 2, \dots, 20\}$ , and the penalty coefficient  $C$  is set to  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$  for experimentation. The experimental results show that accuracy is increased with the increase in the number of receptive fields, and accuracy is slightly reduced with the increase in the penalty coefficient. The accuracy of the combined structure can reach 95.2% when the number of receptive fields per layer is 2. When the number of receptive fields is increased to 12, the accuracy is all over 99%. When the number of receptive fields is 38 and the penalty coefficient is 0.001, the accuracy of ELM-ARF reaches 99.74%, which is the highest in Fig. 6. This shows the effectiveness of the combination of  $4 \times 4$  and  $3 \times 3$ .

In Table 1, the accuracy of the ELM-ARF is compared to other algorithms. For fair comparison, the penalty coefficient for all algorithms is selected from  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . ELM-ARF is set to 2 layers. The total number of receptive fields is 24. Each convolutional layer contains 6 local receptive fields and 6 global receptive fields. And the penalty coefficient is 0.1. ELM-LRF is

set to single layer, 24 local receptive fields of  $4 \times 4$  size and pooling size of  $3 \times 3$ . CKELM is set to two layers, 24 local receptive fields of  $8 \times 8$  size and pooling size of  $3 \times 3$ . The DC-ELM is consistent with the literature [18]. The ELM-MSLRF is set to single layer with 24 local receptive fields, and the size of local receptive field and pooling is set according to the literature [20]. Other parameters are consistent with ELM-ARF. Among the several algorithms in Table 1, ELM-ARF achieves the highest classification accuracy in the case of low training time. This proves the effectiveness of ELM-ARF on small database. In this paper, the highest testing accuracy in each table is shown in bold.

### 4.2 MNIST database

In order to test the classification ability of ELM-ARF on a database with simple image content and large number of images, MNIST is selected for experiments. Example images in the database are shown in Fig. 7. The MNIST database contains 70,000 handwritten digital grayscale images from 0 to 9, of which 60,000 are used as training samples and 10,000 are used as test samples. Each image is size-normalized to  $28 \times 28$  pixels, and the content is centered. We use 60,000 images for training and 10,000 for testing.

Figure 8 shows the accuracy mesh of ELM-ARF with different parameter combinations on MNIST, and its arrangement is consistent with Fig. 5. When the size of local receptive fields is  $4 \times 4$ , the accuracy of the 3 layers



Fig. 7 Example images in the MNIST database

is lower than that of the 2 layers. Considering that the combination of  $4 \times 4$  and  $3 \times 3$  has achieved good results on the USPS database, we continue to experiment with the same combination as in Sect. 4.1. The parameter  $g$  is set to  $\{1, 2, \dots, 15\}$ , and the penalty coefficient is set to  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . The experimental results are shown in Fig. 9. Figure 9a, b is the mesh diagrams of accuracy and cost time of the combination of  $4 \times 4$  and  $3 \times 3$ . When 30 ( $g = 15$ ) receptive fields per layer are used, the highest accuracy of 98.95% can be achieved in Fig. 9a. Therefore, the combination of  $4 \times 4$  and  $3 \times 3$  is more effective on the MNIST database.

Figures 5, 6, 8 and 9 show that accuracy is increased with the increase in  $g$ , and the change in  $C$  has a little effect on accuracy. Comparing the mesh diagrams of the same column in Figs. 5 and 8, it is observed that the accuracy of ELM-ARF with 2 layers is generally higher than that of the 1 layer, but the addition of the third layer has no significant

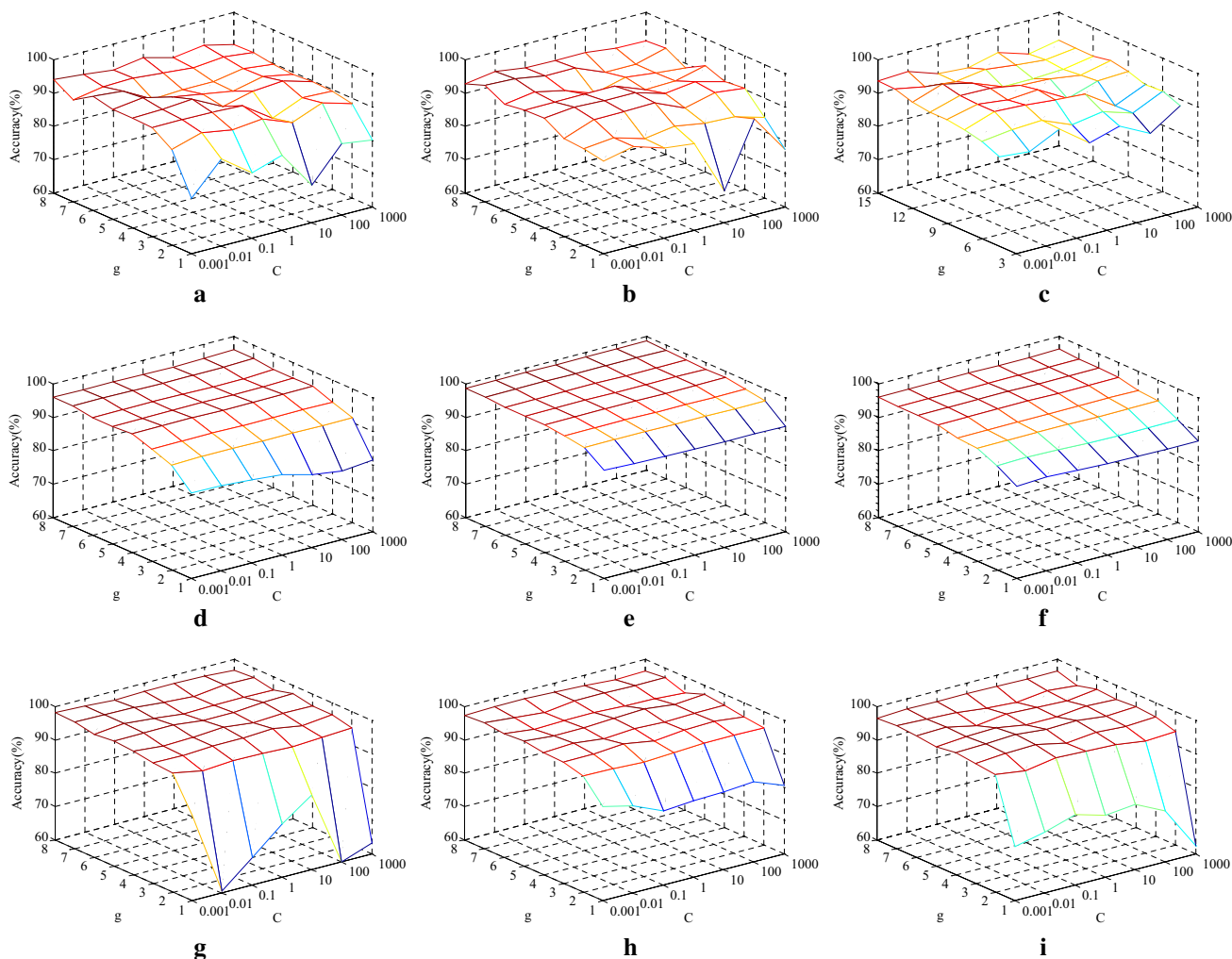
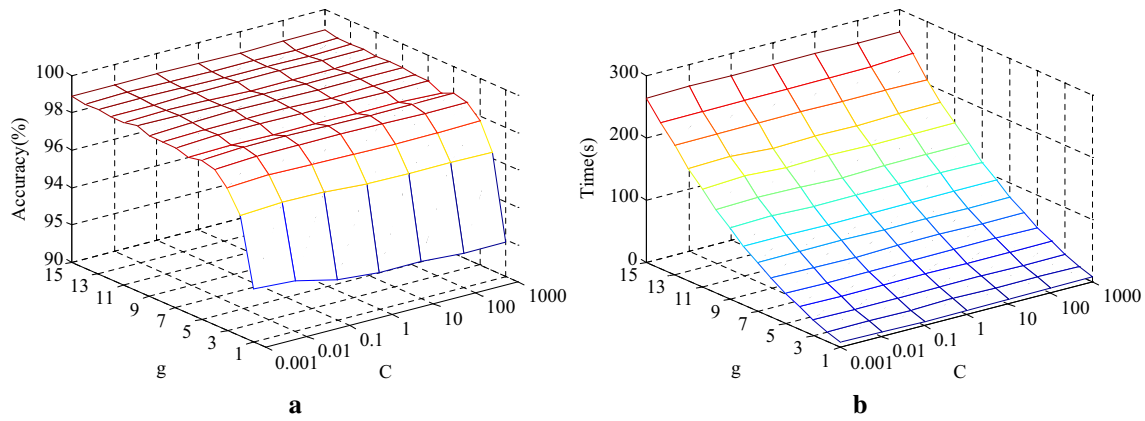


Fig. 8 ELM-ARF parameters selection on MNIST database. **a** 1 layer and  $3 \times 3$  LRF, **b** 1 layer and  $4 \times 4$  LRF, **c** 1 layer and  $5 \times 5$ , **d** 2 layers and  $3 \times 3$ , **e** 2 layers and  $4 \times 4$ , **f** 2 layers and  $5 \times 5$ , **g** 3 layers and  $3 \times 3$ , **h** 3 layers and  $4 \times 4$ , **i** 3 layers and  $5 \times 5$



**Fig. 9** Accuracy and time of ELM-ARF (2 layers,  $4 \times 4$  and  $3 \times 3$ ) on MNIST database

**Table 2** Comparison of accuracies and time on MNIST database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
ELM [20]	431.8	–	97.39
XCov-NCAE [28]	1458.77	$5.5e-6$	97.42
ELM-LRF [20]	500.2	–	97.59
ELM-MSLRF [20]	531	–	98.57
SAE [29]	68,400	–	98.6
CAE-ELM [29]	1090	–	98.87
DBN [29]	20,520	–	98.87
ELM-ARF	265	22	<b>98.95</b>

effect on the improvement of accuracy. Comparing the mesh diagrams of the same row, it can be found that the increase in the size of the local receptive field has few effects on the accuracy. Therefore, when classifying digital images,  $g$  and the number of layers have an important influence on the classification performance of ELM-ARF. When the number of layers is 2 and  $g < 7$ , the accuracy increases significantly as  $g$  increases, and higher accuracy can be obtained in less time. When  $g > 7$ , the effect of accuracy being improved is reduced, but the accuracy continues to be improved.

Experimental comparison with some algorithms using the same number of samples is shown in Table 2, in which experimental results published in other studies are listed. The network parameters used by each algorithm in Table 2 are different. For example, ELM-LRF is set to a single layer and 48 local receptive fields, and ELM-MSLRF even uses 200 local receptive fields. ELM-ARF achieves the highest accuracy in Table 2 with only 30 receptive fields per layer.

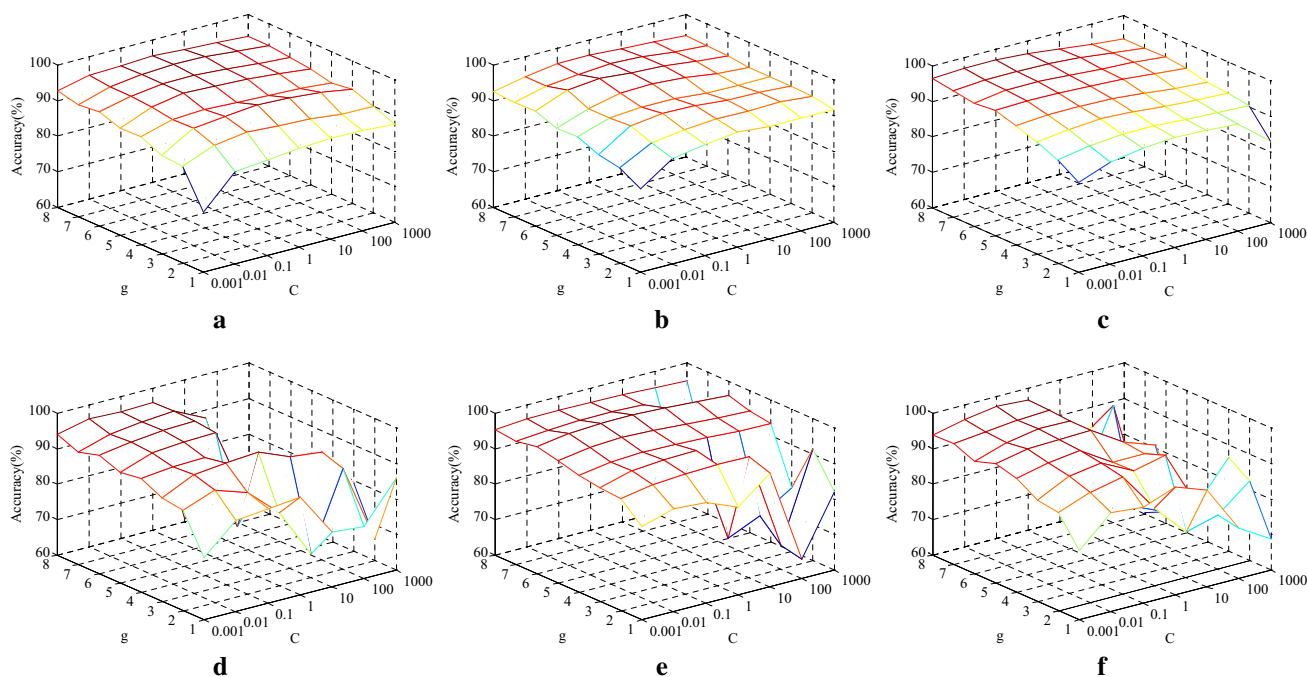
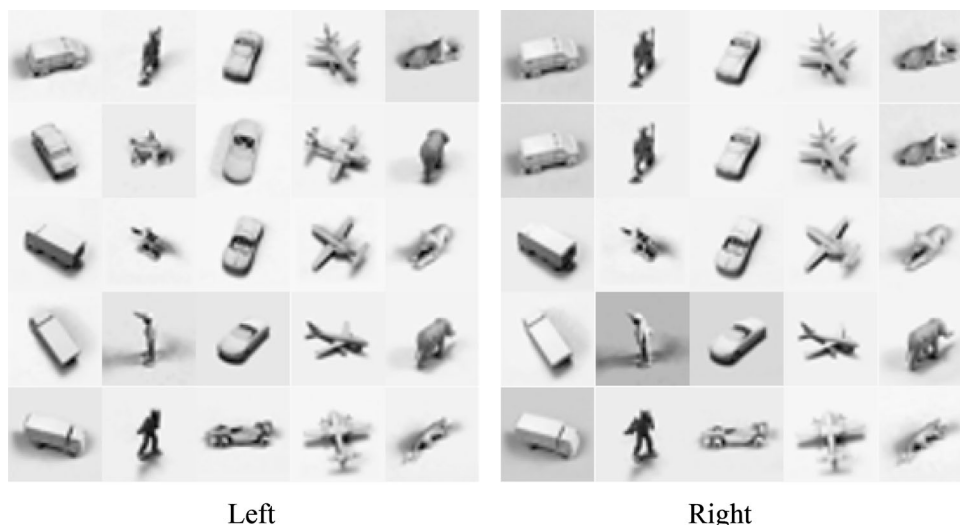
In order to reduce the amount of calculation, some algorithms randomly select 10,000 or 15,000 samples from 60,000 samples for training, such as CKELM [17] and DC-ELM [18]. For fair comparison, we use 60,000 samples for training and 10,000 for testing. The network parameters of

these algorithms are set to be the same as ELM-ARF, and the experimental results are shown in Table 3. The penalty coefficient for all algorithms is selected from  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . The number of layers of ELM-ARF is set to 2. The total number of receptive fields is 20, and the penalty coefficient is 1000. The parameters of the DC-ELM are set to be consistent with the literature [18]. ELM-LRF is set to a single layer, 20 local receptive fields of  $4 \times 4$  size and pooling size of  $3 \times 3$ . The ELM-MSLRF is set to a single layer with 20 local receptive fields, and the size of local receptive field and pooling is set according to the literature [20]. CKELM is set to 2 layers, 20 local

**Table 3** Comparison of unified experimental environment and parameters on MNIST database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
ELM-LRF	47	4.6	97.9
CKELM	33	4.6	97.64
DC-ELM	65	9.3	98.3
ELM-MSLRF	46	5	97.8
ELM-ARF	48	5.5	<b>98.63</b>

**Fig. 10** Example images (left and right sides) in the NORB database



**Fig. 11** ELM-ARF parameters selection on NORB database. **a** 2 layers and  $3 \times 3$  LRF, **b** 2 layers and  $4 \times 4$  LRF, **c** 2 layers and  $5 \times 5$  LRF, **d** 3 layers and  $3 \times 3$  LRF, **e** 3 layers and  $4 \times 4$  LRF, **f** 3 layers and  $5 \times 5$  LRF

receptive fields of  $5 \times 5$  size and pooling size of  $2 \times 2$ . Under the same calculation conditions, the training time of ELM-ARF is relatively long, which is caused by calculating the global receptive field and the feature coding, but the accuracy is the highest in Table 3. From the experimental results in Sects. 4.1 and 4.2, it can be concluded that ELM-ARF has a good classification effect on handwritten digital images when the local receptive field size of the first layer is set to  $4 \times 4$  and the size of the second layer is set to  $3 \times 3$ .

### 4.3 NORB database

The images in USPS and MNIST are digital images, and the image content is simple. To test the ability of ELM-ARF to process images of complex content, we used the NORB database for experiment. The database contains a total of five categories of toy objects: people, animals, airplanes, trucks and cars. Each category contains 10 instances, and the database has a total of 50 instances. By utilizing different viewpoints and various lighting conditions, each instance contains 972 stereoscopic images.

Each stereo image has 2 images (left and right sides). All images are unified into  $32 \times 32$  pixels. Example images in the database are shown in Fig. 10. During the experiment, we select 5 instances in each category and use a total of 24,300 image pairs for training. We select 5 remaining instances in each category and use a total of 24,300 image pairs for testing.

On the USPS and MNIST databases, the classification accuracy of ELM-ARF with 2 layers is better than that of 1 layer. We carry out experiment by setting the number of layers to 2 and 3 on the NORB database. The other parameter settings are the same as 4.1 and 4.2. In the mesh diagrams of the first row, the change trend of accuracy is relatively stable when the number of receptive fields and the penalty coefficient change. In the mesh diagrams of the second row of Fig. 11, the accuracy of 3 layers is obviously oscillated with the increase in the penalty coefficient, which decreases the average accuracy. In Fig. 11b, the  $4 \times 4$  receptive field reaches the highest accuracy of 96.7% in the three mesh diagrams when there are only 12 receptive

fields per layer and the penalty coefficient is 0.1. We also use the combination of  $4 \times 4$  and  $3 \times 3$  to perform experiments, and their accuracy is lower than that of  $4 \times 4$  local receptive field in Fig. 11b. Therefore, in the NORB database experiment, the number of layers is set to 2, and the size of the local receptive field is set to  $4 \times 4$ .

The ELM-ARF is set to 2 layers, each of which has 6 local receptive fields of  $4 \times 4$  size and 6 global receptive fields, and the penalty factor is 0.1. The images in Fig. 12 are output feature maps of each convolutional layer and pooling layer when the ELM-ARF processes the input image. In the first row, the six images on the left are the feature maps generated by the local receptive fields of the first convolutional layer. The right side is the feature maps generated by the global receptive fields. The images of the second row are pooling maps. The left and right sides are, respectively, generated by pooling the receptive field features. The images in the third row are the feature map generated by the second convolutional layer. The images in the fourth row are pooled maps. Comparing the left and

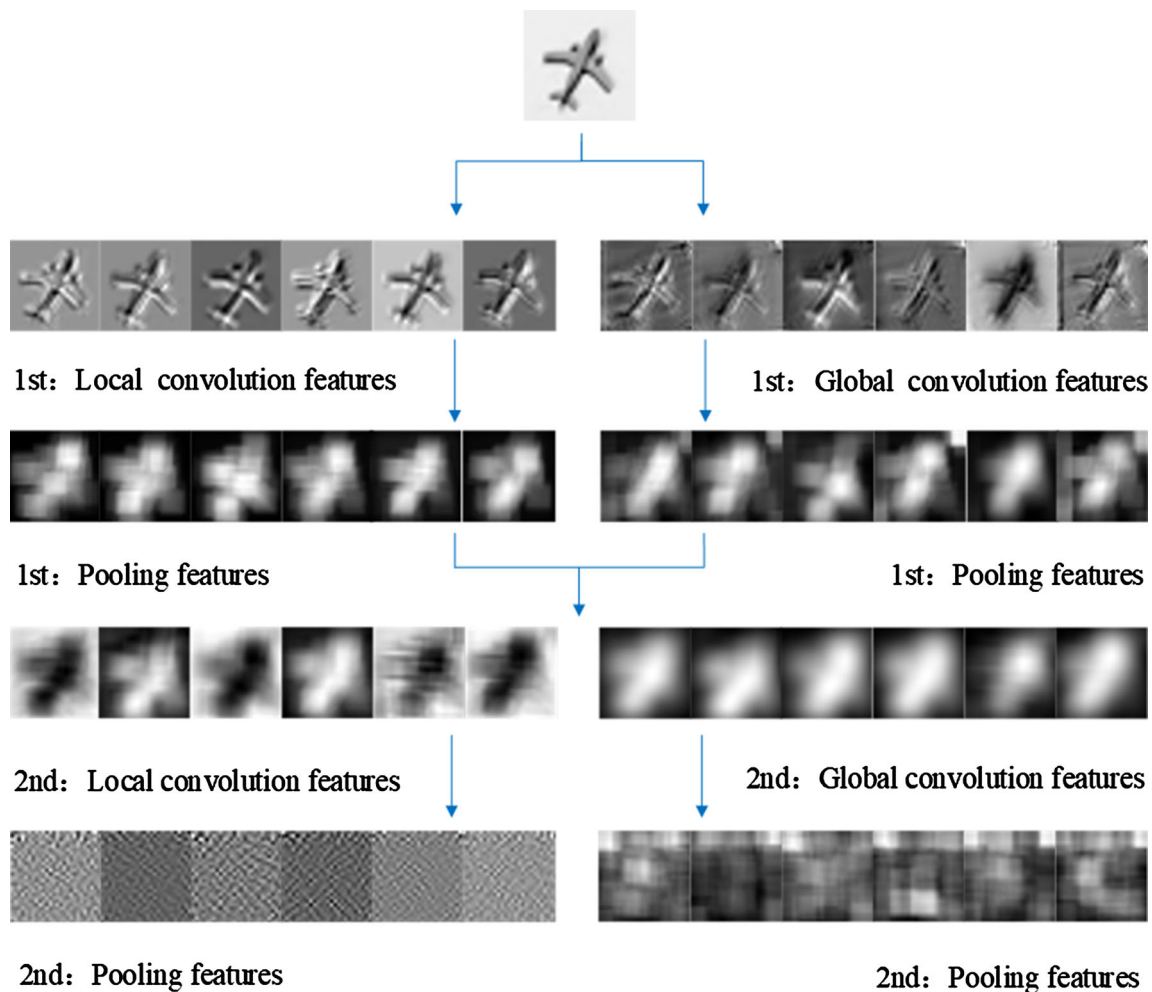
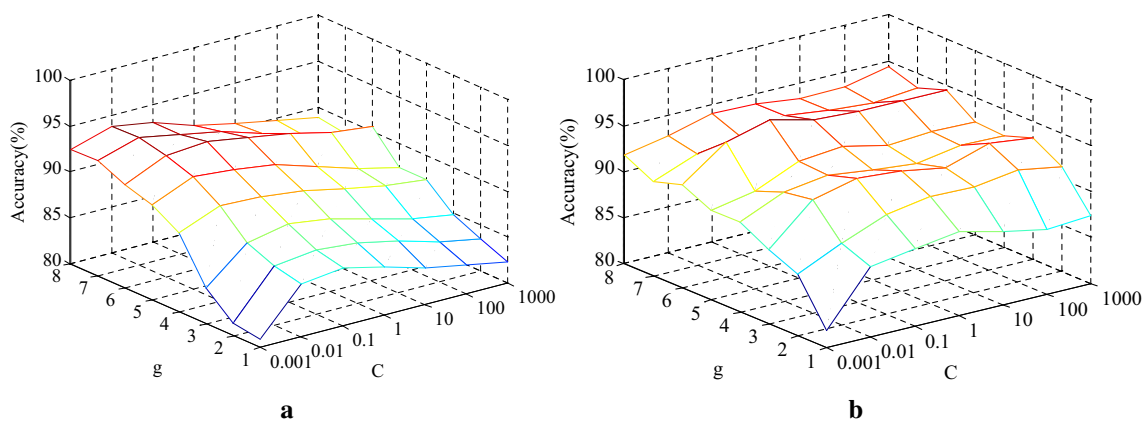


Fig. 12 Output feature maps of ELM-ARF on NORB database



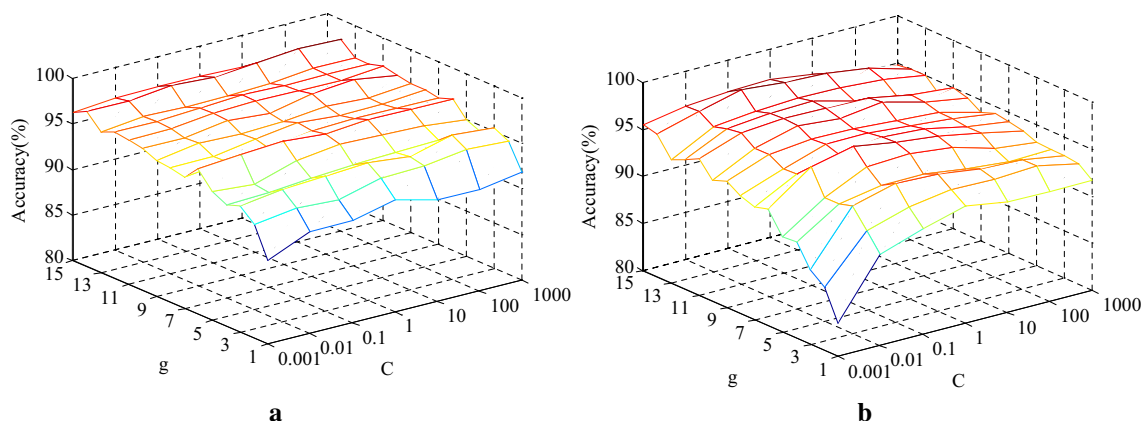
**Fig. 13** Effectiveness of ELM-ARF. **a** Remove the global experience field and **b** remove the feature coding structure

right sides of Fig. 12, the feature maps of the local receptive fields have more texture details, while the feature maps of the global receptive fields have a smoother object contour.

To prove that adding the global receptive fields and the feature coding structure effectively improves the classification performance, experiments are carried out in two aspects: (1) The global receptive fields is replaced by local receptive fields. The experimental results are shown in Fig. 13a. When  $C < 0.1$ , the accuracy increases as  $C$  increases. When  $C > 0.1$ , the accuracy decreases as  $C$  increases. The highest accuracy is only 93.8% when  $g = 7$ , and the training takes 47.3 s. (2) The feature coding structure is removed, and the output features of the second layer are directly input into the classification layer trained with ELM. The experimental results are shown in Fig. 13b. The accuracy increases as  $C$  increases. The highest accuracy is only 95.3% when  $g = 6$ , and the training takes 42.4 s. Compared with Fig. 11, the accuracy of the above two experiments is decreased. In Fig. 11b, the highest accuracy is 96.7% when  $g = 6$ , and the training takes 49.3 s. This proves that the combination of global receptive

fields and feature coding structure effectively improves the accuracy when the training time is slightly increased.

The trends of accuracy change with respect to  $C$  in Fig. 13a, b are different. Figure 13a, b contains local receptive fields, so the accuracy difference between them is caused by the global receptive fields and the feature coding structure. It can be found that the accuracy of the structure with the global receptive fields is not reduced as the  $C$  increases, and the accuracy of the structure with feature coding reaches the highest at  $C = 0.1$ . Therefore, we combine the setting methods of the penalty coefficient in the above two structures and experiment. The penalty coefficient  $C$  of the feature coding structure is set to 0.1, and the penalty coefficient of the global receptive field is set to  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . The experimental results are shown in Fig. 14a. We also set the two penalty coefficients  $C$  uniformly to  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$  for experimentation. The experimental result is shown in Fig. 14b. Compared to Fig. 14b, a has a more stable growth trend and higher accuracy. The coefficient  $C$  of the feature coding structure has defaulted to 0.1 in Fig. 14a. ELM-ARF achieves the highest accuracy when the  $C$  of the global receptive fields



**Fig. 14** Penalty coefficient  $C$  selection, **a**  $C$  of the feature coding structure are set to 0.1, **b**  $C$  of the global receptive fields and feature coding structure are uniformly changed

**Table 4** Comparison of accuracies and time on NORB database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
XCov-NCAE [28]	609	1.3e−5	82.33
SAE [29]	85,717	–	93.5
DBN [29]	15,104	–	92.8
ML-EKM-ELM [30]	60.33	11.62	93.17
CAE-ELM [29]	1208	–	94.5
ELM-LRF [21]	397	–	97.26
ELM-MSLRF [21]	403	–	97.5
ELM-HLRF [21]	516	–	97.45
ELM-ARF	216	68.2	<b>98</b>

**Table 5** Comparison of unified experimental environment and parameters on NORB database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
ELM-LRF	262	46	97
CKELM	114	21	93.3
DC-ELM	46	34	89
ELM-MSLRF	236	47	97.1
ELM-ARF	219	69	<b>98</b>

is 100 and  $g = 14$ . The highest accuracy in Fig. 14a is 98%, which can prove the effectiveness of the global receptive fields.

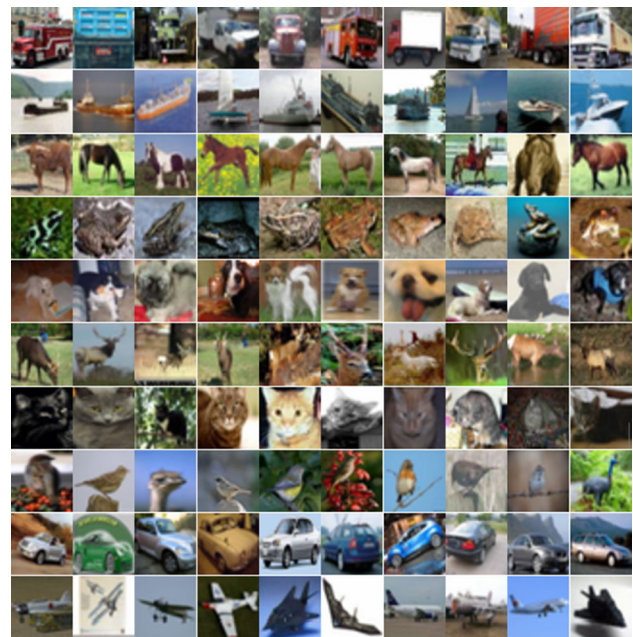
In Figs. 11, 13 and 14, the increase in  $g$  and  $C$  has an important influence on accuracy. Comparing the mesh diagrams of the same row in Fig. 11, it can be found that the increase in the size of the local receptive field has few effects on the accuracy. Comparing the mesh diagrams of the same column, it can be found that the increase in the number of layers causes large fluctuations in accuracy. Therefore, when classifying object images,  $g$  and  $C$  have an important influence on the classification performance of ELM-ARF. When  $C$  of the feature coding structure is less than 1, the accuracy is improved as  $g$  and  $C$  of the global receptive fields increase within the given range.

The experimental results published in some studies are listed in Table 4. The number of layers of ELM-ARF is set to 2 layers in which the total number of receptive fields is 56. The  $C$  of the feature coding structure is 0.1, and the  $C$  of the global receptive fields is 100. ELM-ARF achieves the highest accuracy of 98% in Table 4. For fair comparison, the penalty factor for all algorithms is selected from  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . ELM-LRF is set to a single layer, 56 local receptive fields of size  $4 \times 4$  and pooling size of  $3 \times 3$ . The ELM-MSLRF is set to a single layer with 56 local receptive fields, and the size of local receptive fields and pooling is set according to the literature [20]. Two-layer CKELM is set to 56 local receptive fields of  $4 \times 4$  size and pooling size of  $3 \times 3$ . The setting of DC-ELM is consistent with that in [18]. The experimental results are shown in Table 5. When the total number of receptive

fields is the same, ELM-ARF obtains the highest accuracy with the third fastest training speed, which shows the effectiveness of the framework in dealing with complex image classification problems.

#### 4.4 CIFAR10 database

Finally, ELM-ARF is used to challenge the object image database CIFAR10 commonly used for deep learning. The

**Fig. 15** Example images in the CIFAR10 database

**Table 6** Comparison of accuracies and time on CIFAR10 database

Methods	Training time (s)	Testing time (s)	Testing accuracy (%)
ELM-LRF	259	16	53.2
CKELM	137	20	55
DC-ELM	86	15	51.92
ELM-MSLRF	249	17	52.7
ELM-ARF	247	22.7	<b>60.14</b>

CIFAR10 data set [26] contains 10 categories of color images. The size of image is  $32 \times 32 \times 3$ . Example images in the database are shown in Fig. 15. The training set contains 50,000 images, and the testing set contains 10,000 images. All images are used for training and testing.

None of the algorithms in Table 6 are tested on this database, so these algorithms are trained using the same conditions as ELM-ARF. The ELM-ARF is set to 2 layers in which the number of receptive fields is 44. The coefficient  $C$  of the feature coding structure is set to 0.001, and the coefficient  $C$  of the global receptive fields is set to 1000. The penalty coefficient for other algorithms is selected from  $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ . The total number of local receptive fields of ELM-LRF, CKELM and ELM-MSLRF is set to be the same as ELM-ARF. The setting of DC-ELM is consistent with that in [18]. ELM-ARF obtains the highest accuracy with the third fastest training speed, and the accuracy of ELM-ARF is 5% higher than that of the second highest CKELM.

## 5 Conclusion

In this paper, the extreme learning machine with autoencoding receptive fields (ELM-ARF) is proposed to effectively utilize features with global attributes of the image and the features extracted by each layer. By using the theory of ELM-AE to train the global receptive fields, ELM-ARF can effectively avoid instability caused by the random initialization of the receptive field matrix while extracting the global features. At the same time, the shallow layer features can be input to any deep layer to be combined through the structure of the ELM-ARF, so that the feature of each layer from shallow to deep is effectively utilized. The experimental results show that ELM-ARF can achieve higher accuracy on the above four databases with less speed reduction. It can be proved that ELM-ARF can effectively deal with object classification problems.

Although the classification results of ELM-ARF on CIFAR10 cannot be compared with CNN, ELM-ARF does not use the time-consuming reverse iteration to train the network. By utilizing the theory of ELM-LRF and ELM-AE, 63% of the CNN classification accuracy is achieved with only 247 s. In future research, the local receptive field

will be further studied to improve ELM-ARF while dealing with classification problems.

**Acknowledgments** This work is supported by National Natural Science Foundation of China (No. 51641609).

## Compliance with ethical standards

**Conflict of interest** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

- Huang GB, Chen L (2007) Convex incremental extreme learning machine. *Neurocomputing* 70(16–18):3056–3062
- Huang GB, Chen L (2008) Enhanced random search based incremental extreme learning machine. *Neurocomputing* 71(16–18):3460–3468
- Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural Netw* 2:985–990
- Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70(1–3):489–501
- Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern Part B Cybern* 42(2):513–529
- Koniusz P, Yan F, Gosselin PH, Mikolajczyk K (2017) Higher-order occurrence pooling for bags-of-words: visual concept detection. *IEEE Trans Pattern Anal Mach Intell* 39(2):313–326
- Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *IEEE Comput Soc Conf Comput Vis Pattern Recognit* 2:2169–2178
- Han HG, Wang LD, Qiao JF (2014) Hierarchical extreme learning machine for feedforward neural network. *Neurocomputing* 128:128–135
- Li G, Niu P, Duan X, Zhang X (2014) Fast learning network: a novel artificial neural network with a fast learning speed. *Neural Comput Appl* 24(7–8):1683–1695
- Qu BY, Lang B, Liang JJ, Qin AK, Crisalle OD (2016) Two-hidden-layer extreme learning machine for regression and classification. *Neurocomputing* 175:826–834
- Huang GB, Bai Z, Kasun LLC, Vong CM (2015) Local receptive fields based extreme learning machine. *IEEE Comput Intell Mag* 10(2):18–29
- Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. *Proc IEEE Conf Comput Vis Pattern Recognit* 2:4700–4708
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105



14. Zhang H, Cao X, Ho JK, Chow TW (2017) Object-level video advertising: an optimization framework. *IEEE Trans Industr Inf* 13(2):520–531
15. Zhang H, Ji Y, Huang W, Liu L (2018) Sitcom-star-based clothing retrieval for video advertising: a deep learning framework. In: *Neural computing and applications*, pp 1–20
16. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Cogn Model* 5(3):1
17. Ding S, Guo L, Hou Y (2017) Extreme learning machine with kernel model based on deep learning. *Neural Comput Appl* 28(8):1975–1984
18. Pang S, Yang X (2016) Deep convolutional extreme learning machine and its application in handwritten digit classification. In: *Computational intelligence and neuroscience*, pp 1–10
19. Liu H, Li F, Xu X, Sun F (2018) Multi-modal local receptive field extreme learning machine for object recognition. *Neurocomputing* 277:4–11
20. Huang J, Yu ZL, Cai Z, Gu Z, Cai Z, Gao W, Yu S, Du Q (2017) Extreme learning machine with multi-scale local receptive fields for texture classification. *Multidimens Syst Signal Process* 28(3):995–1011
21. He B, Song Y, Zhu Y, Sha Q, Shen Y, Yan T, Nian R, Lendasse A (2018) Local receptive fields based extreme learning machine with hybrid filter kernels for image classification. In: *Multidimensional systems and signal processing*, pp 1–21
22. Kasun LLC, Zhou H, Huang GB, Vong CM (2013) Representational learning with extreme learning machine for big data. *IEEE Intell Syst* 28(6):31–34
23. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
24. Lu J, Lu Y, Cong G (2011) Reverse spatial and textual k nearest neighbor search. In: *Proceedings of the 2011 ACM SIGMOD international conference on management of data*. ACM, pp 349–360
25. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
26. LeCun Y, Huang FJ, Bottou L (2004) Learning methods for generic object recognition with invariance to pose and lighting. In: *Proceedings of the computer vision and pattern recognition*, pp 97–104
27. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Technical report, University of Toronto, vol 1, p 7
28. Chen J, Wu Z, Zhang J, Li F, Li W, Wu Z (2018) Cross-covariance regularized autoencoders for nonredundant sparse feature representation. *Neurocomputing* 316:49–58
29. Wang Y, Xie Z, Xu K, Dou Y, Lei Y (2016) An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning. *Neurocomputing* 174:988–998
30. Vong CM, Chen C, Wong PK (2018) Empirical kernel map-based multilayer extreme learning machines for representation learning. *Neurocomputing* 310:265–276

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.