Neural Comput & Applic
DOI 10.1007/s00521-015-2008-7

ORIGINAL ARTICLE

Local extreme learning machine: local classification model for shape feature extraction

Jing Zhang[1] · Lin Feng[1] · Bin Wu[2]

Received: 30 June 2014 / Accepted: 9 August 2015

Abstract The shape feature of an object represents the geometrical information which plays an important role in the image understanding and image retrieval. How to get an excellent shape feature that has rotation, scaling and translation (RST) invariance is a problem in this field. This paper proposed a novel local extreme learning machine (LELM) classification algorithm to extract the shape features. LELM finds nearest neighbors of the testing set from the original training set and trains a local classification model. The shape feature is represented by an analytic decision function with a radial basis function (RBF) kernel obtained by LELM. Our method has the following advantages: (1) LELM not only keeps the local structure of the samples, but also solves the imbalance problem between variance and bias. (2) Features obtained by the RBF kernel are RST invariant. (3) LELM is more robust against the noise and fragmentation compared to other methods. We also demonstrate the performance of the proposed method in image retrieval.

Keywords Extreme learning machine (ELM) · Local extreme learning machine (LELM) · Feature point · Image retrieval

✉ Lin Feng
fenglin@dlut.edu.cn

1 Faculty of Electronic Information and Electrical Engineering, School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China

2 School of Innovation Experiment, Dalian University of Technology, Dalian 116024, Liaoning, China

1 Introduction

As one of the most important parts of big data, image data contains rich information, such as color, texture, shape [1]. Shape is one of the most important features in image and plays a key role in the understanding of target. It provides a premise for image comparison, identification, classification, and is widely used in image retrieval. MPEG-7 standard suggests that the shape features should have strong ability of description [2]. The extracted features can distinguish class which different object shapes belong to. Also, the features have rotation, scaling and translation (RST) invariance, and have robustness for noise interference, shape distortion, local defects. Currently, many researchers have made contributions to the shape features representation and description, which can be summarized as two classes, one is region-based and the other is contour-based [3]. Human mainly distinguishes the shape through the object contour. Many contour representations require small amount of calculation, which is suitable for complex image processing and computer vision applications. Contour representation becomes the hot topic. The description of shape is based on contour includes both global and local methods.

Local algorithm uses broken line approximation, curvature, curve fitting to divide the shape into several areas, and then it generates feature description. Mokhtarian et al. come up with curvature scale space algorithm (CSS) that has translation and scaling invariance [4–6]. However, this algorithm also has some problems, for instance, a nonexistent curvature convex shape of a zero curvature (such as circle). The initial position of sampling points directly influences the outcome of shape matching. Lien and Liu et al. make an improvement on this algorithm for popularizing its application [7, 8], which makes it widely used in the field of shape analysis. In order to solve the problems of the local concave and convex shape sensitivity, Adamek and Connor come up with a multi-scale space description method which uses convex and concave degree of shape to describe shape features, called multi-scale convexity concavity (MCC) [9]. This method has the RST invariance and is not sensitive to slight deformation. To show the internal structure of more complex shapes, Ling et al. put forward the internal distance description method in 2007 [10], which has low computational complexity and has better effect than Euclidean distance on images with articulated shape. But this algorithm is sensitive to covered image. Random walk [11] and geodesic distance [12] are also used in the local description of contour. Although the local structure algorithm is robust for noise, it is sensitive to images which have internal or external crack.

Global structure algorithm uses multidimensional feature vector to show the information of the shape edge. Then, it compares the similarity of shape by computing the distance (e.g., Euclidean distance), including calculus of variations [13], the level integrated [14, 15], among which the shape description based on feature points is outstanding. Belongie et al. put forward shape context (SC) which is based on the correspondence of sampling points of contour to conduct shape matching and recognition in 2004 [16, 17]. SC algorithm which is essentially of the RST invariance has certain ability against deformation. These advantages make SC widely used in the field of shape analysis. The above-mentioned algorithms required relatively small amount when they are used to extract features, which makes them fitter for the shape extraction of complex images. However, they are only good at describing single target and sensitive to noise and deformation. Van Nguyen et al. come up with shape representation method called SVS which treats image shape description with a new view that regards shape extraction process as a binary classification problem [18]. This algorithm is based on support vector machine (SVM) classification theory [19, 20], extracts feature points of shape by training SVM classifier and then conducts a census for the point surrounding gradient histogram to obtain the feature vector. This algorithm not only extracts the shape feature of images, but also keeps the RST invariance by using RBF kernel function. It also keeps the internal feature of images well and has the ability against noise and external disturbance. But this method needs to convert image data to binary classification training data, which has large size and high dimension of training data. Besides, the way to solve SVM classification problem has following problems: Iteration process of looking for support vectors slows down the computation speed; it is difficult to get a bigger solution space, for solving the optimization problem needs considering equality constraint; it belongs to the global optimization model and is hard to establish a classification model for all the training data of high dimension.

To solve the problems generated by SVM, Huang et al. come up with extreme learning machine (ELM) which is single-hidden-layer feed-forward neural networks (SLFNs) [21]. The essence of ELM is that the hidden layer of SLFNs need not be tuned. Compared with those traditional computational intelligence techniques, ELM provides better generalization performance at a much faster learning speed, and it avoids the results converging into a local minimum. Zong et al. applied the algorithm on face recognition and have achieved good result [22]. In order to extend application of ELM algorithm, many researchers have made improvements on it [23–25]. Meanwhile, Huang et al. conduct experimental verification and have theory proved that ELM does better in solving high-dimensional data classification than SVM [26, 27]: ELM need not consider the selection of the parameters, ELM need not consider the optimal solution of equality constraint problems, and ELM has a bigger solution space.

Both ELM and SVM try to find classification boundaries that are proper for all of the training data through solving a global optimization problem. And this way will generate follow problems:

- Both classification models are only for the global structure, but do not take into account the local structure.
- All training samples for computing increase the computational complexity.

To solve the problems, this paper proposes local extreme learning machine (LELM) model which belongs to local optimization model. This model reconstructs the new training set by calculating the distance to find close neighbors of the testing set, uses the new training set to train LELM classifier as a decision function to extract feature points of the shape and makes a statistics for gradient histogram around feature points to get image characteristics express for image retrieval. Experimental validation of the method has a remarkable improvement in image retrieval applications.

This paper is organized as follows: the second section reviews ELM algorithm and compares with SVM when the same conditions are used; the third section states the shortages of ELM and puts forward LELM algorithm; the fourth section shows how LELM algorithm extracts feature points and applies it in image retrieval; the fifth section checks the effect of our algorithm by experiments.

2 Preliminary knowledge

2.1 A brief review of ELM

Extreme learning machine is an improved single-hidden-layer neural network. SLFNs have problems: the solving speed is slow, and it is easy to be trapped in local minimum. It can be described as follows: for N arbitrary distinct samples (x_i, t_i), where $X = [x_1, x_2, \ldots, x_N]^T \in \mathbb{R}^{D \times N} t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathbb{R}^m$, it is used to solve multi-classification problems, the number of network output nodes are $m (m \geq 2)$. There are $\widetilde{N}$ hidden layer nodes in networks, and activation function $g(\cdot)$ can be Sigmoid or RBF:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(a_i x_j + b_j) = o_j$$

where $j = 1, \ldots, N$, $a_i = [a_{i1}, a_{i2}, \ldots, a_{in}]^T$ is the input weight vector which connects the ith hidden layer notes with input notes, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the output weight vector which connects hidden notes with output notes, and b_i is bias value of i hidden layer notes.

The number of hidden notes is $\widetilde{N}$, written in matrix form: $H\beta = T$, where the hidden layer of network output matrix is:

$$H = \begin{pmatrix} g(a_1 x_1 + b_1) & \cdots & g(a_{\widetilde{N}} x_1 + b_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ g(a_1 x_N + b_1) & \cdots & g(a_{\widetilde{N}} x_{\widetilde{N}} + b_{\widetilde{N}}) \end{pmatrix}_{N \times \widetilde{N}},$$

$$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{pmatrix}_{\widetilde{N} \times m}, T = \begin{pmatrix} t_1^T \\ \vdots \\ t_N^T \end{pmatrix}_{N \times m}$$

Through gradient descent, the traditional SLFNs get values of $\hat{a}_i, \hat{b}_i, \hat{\beta}_i (i = 1, \ldots, \widetilde{N})$ subjecting to:

$$\| H\left(\hat{a}_1, \ldots, \hat{a}_{\widetilde{N}}, \hat{b}_1, \ldots, \hat{b}_{\widetilde{N}}\right)\beta - T \| \\ = \min \| H\left(a_1, \ldots, a_{\widetilde{N}}, b_1, \ldots, b_{\widetilde{N}}\right)\beta - T \|$$

The input weight a_i and the bias value of hidden layer b_i can be generated randomly, which is known by [21]. The solution of SLFNs can be obtained by the least square of linear system $H\beta = T$. The solution form of $H\beta = T$ can be written as: $\hat{\beta} = H^+ T$, where H^+ is the generalized inverse matrix of H. ELM is minimizing both the training errors and the output weights. It can express arbitrary continuous objective function. And ELM classifier can be infinite close to its objective function. The expression can be gotten based on optimization of the ELM:

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

$$\text{Subject to}: t_i \beta \cdot h(x_i) \geq 1 - \xi_i, \quad i = 1, \ldots, N \tag{1}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, N$$

where $\xi_i = \begin{pmatrix} \xi_{i,1} & \cdots & \xi_{i,m} \end{pmatrix}$ is the vector of the training errors. Based on KKT theory, and by Lagrange multiplier, the optimal function can be converted as following:

$$\text{Minimize}: L_{\text{ELM}} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t_i t_j \alpha_i \alpha_j x_i x_j - \sum_{i=1}^{N} \alpha_i \\ \text{Subject to}: 0 \leq \alpha_i \leq C, i = 1, \ldots, N \tag{2}$$

where $\alpha = \begin{pmatrix} \alpha_i & \cdots & \alpha_N \end{pmatrix}^T$, and:

$$\beta = H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \tag{3}$$

The output function of ELM is:

$$f(x) = h(x)\beta = h(x)H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \tag{4}$$

The matrix expression of ELM kernel function can be defined as:

$$\Omega_{\text{ELM}} = HH^T : \Omega ELM_{i,j} = h(x_i) \cdot h(y_j) = K(x_i, x_j) \tag{5}$$

ELM output function can be obtained from (4) and (5), where (4) can be shown as kernel form:

$$f(x) = \begin{pmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{pmatrix} \left(\frac{I}{C} + \Omega\text{ELM}\right)^{-1} T \tag{6}$$

Here, the proper kernel function can be selected, such as RBF kernel function $K(u, v) = exp(-\gamma\|u - v\|^2)$. And $K(x_i, x_j) = h(x_i) \cdot h(x_j)$ is called as ELM kernel. The optimization equation is:

$$\Omega_{\text{ELM}} = HH^T : \Omega\text{ELM}_{i,j} = h(x_i) \cdot h(y_j) = K(x_i, x_j) \tag{7}$$

ELM algorithm can be summarized as:

1. Randomly select the network input weight a_l and bias value $b, l = 1, \ldots \widetilde{N}$;
2. Calculate hidden layer output matrix H;
3. Calculate output weight $\hat{\beta} = H^+ T$, where $T = \begin{pmatrix} t_1 & \ldots & t_N \end{pmatrix}^T$.

2.2 Compared with SVM

The SVM algorithm as an important classification method is described as follows: given a set of training data $(x_i, t_i), i = 1, \ldots, N$, where $x_i \in \mathbb{R}^D$ and $t_i\{-1, 1\}$, SVM aims to maximize the separating margin of the two different classes as well as to minimize the training errors, which is equivalent to:

$$\text{Minimize} : L_{\text{SVM}} = \frac{1}{2}\|w\|^2 + C\sum\nolimits_{i=1}^{N} \xi_i$$

$$\text{Subject to} : t_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N$$

$$\xi_i \geq 0, \quad i = 1, \ldots, N \tag{8}$$

where C is regularization parameter and provides a trade-off between minimizing the training error and maximizing the distance w of the separating margin of the two different classes in the feature space $\phi : x_i \rightarrow \phi(x_i)$ is a nonlinear mapping which maps the training data x_i from the input space to a feature space. The SVM optimization can be written as the Lagrange function:

$$\begin{aligned} &\text{Minimize} : L_{\text{SVM}} \\ &\quad = \frac{1}{2}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{N} t_i t_j \alpha_i \alpha_j \phi(x_i)\phi(x_j) - \sum\nolimits_{i=1}^{N} \alpha_i \\ &\text{Subject to} : \sum\nolimits_{i=1}^{N} t_i \alpha_i = 0 \end{aligned} \tag{9}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N$$

where each Lagrange multiplier α_i corresponds to a training sample (x_i, t_i).

Equations (1) and (8) show that SVM and ELM have similar optimization objective function, but different item constraint. Therefore, two algorithms have many differences: According to the SVM theory, the separating hyperplane can be got without taking advantage of the origin feature space in the SVM and thus a bias b is preferred in the optimization constraints of SVM so that the separating hyperplane can be adjusted accordingly: $w \cdot \phi(x_i) + b = 0$ (cf. Fig. 1a). However, all the parameters of the ELM mapping $h(x)$ are randomly generated, and the ELM mapping $h(x)$ becomes known to users finally. Such ELM feature mapping $h(x)$ includes but is not limited to Sigmoid, RBF and so on. According to the ELM theory, the separating hyperplane tends to taking advantage of the origin feature space in the ELM (cf. Fig. 1b). Thus, the bias b is required neither in the output nodes nor in the ELMs optimization constrains (1).

Figure 1 compares classification principle of ELM and SVM. As shown in Fig. 1a, the classification hyperplane of SVM without the original sample space. In Fig. 1b, the classification hyperplane of ELM utilizes the original sample space.

3 Local ELM

Getting the ELM classification model is a complex global training process and the training errors as well as norm of output weights need to be minimized, there are several questions:

- ELM requires all training samples to participate in operation, and it does not take into account the local structure of samples.
- If all samples are trained to get a nonlinear model, the basic technique is to find an optimal separating hyperplane that is to maximize the classification boundary in high-dimensional feature space. This is difficult to keep the balance between variance and bias.
- Due to using all training samples to fit an effective classification boundary, the computational complexity is increased.

Inspired by KNN algorithm, we propose a more general framework called local extreme learning machine (LELM) to solve the limitations of ELM. The basic idea of the LELM is as follows:

1. Get the K nearest neighbors of each testing sample from all the original training samples.

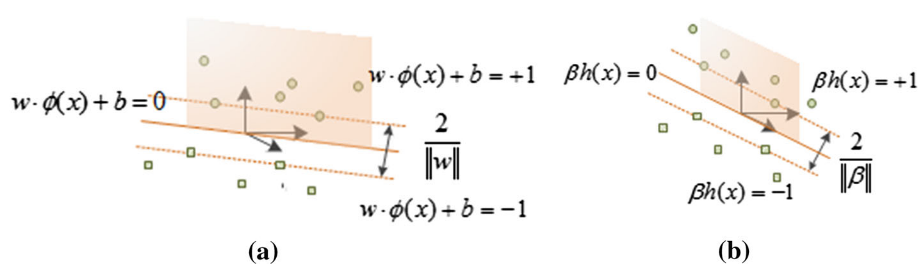

Fig. 1 Comparison of SVM and ELM in classification problem. **a** The feature mappings of SVM are unknown and may not satisfy universal approximation condition, need to be presented to absorb the system error. **b** The feature mappings of ELM are known, and universal approximation capability is considered at the first place. In ELM, the system error tends to be zero and should not be present

2. If the K neighbors have the same label, add the K neighbors to the new training set.
3. Use the new training set to train the local classification model instead of the original global model of all training samples.

The method reduces the impact of outliers on the model. In this paper, the Gaussian function is used to calculate the similarities between the training sample and the testing sample. The function to maximize the ratio of the similar sample and the dissimilar sample increases the degree of similarity distinction between samples. All samples have been normalized into the range [0, 1] before measuring similarities.

We adopt the Gaussian function to measure the similarity which is between the testing sample $\bar{x}_s$ and the training sample x_i:

$$d(\bar{x}_s, x_i) = \exp(\|\bar{x}_s - x_i\|_2^2/\mu) \tag{10}$$

Use "binary function" to select the neighborhood of the testing sample. If the training samples are in the neighborhood (is d) of the testing sample, the "binary function" value is 1, otherwise 0, the "binary function" is as follows:

$$y(d(\bar{x}_s, x_i)) = \begin{cases} 0, & d(\bar{x}_s, x_i) > \sigma \\ 1, & d(\bar{x}_s, x_i) \le \sigma \end{cases} \tag{11}$$

where σ is a user-specified parameter.

According to Sect. 2 described, kernel ELM can be written as (7), where α_i is Lagrange multipliers. The kernel function $K(x_i, x_j)$ can be substituted and can be obtained by $h(x_i) \cdot h(x_j)$ calculation. If the classification model has been trained, then the class labels of the testing samples can be calculated by (6). Therefore, the classification process only depends on the training set; in the other words, the classification process depends on the training samples that correspond to a_i value of nonzero. The similarity between each testing sample and training sample can be calculated by Eq. (10), which is used to adjust the training error of the model. For each $x_i \in \mathbb{R}^N$, we construct a local extreme learning machine model by solving the following optimization problem:

$$\begin{aligned} &\text{Minimize}: L_{\text{LELM}} = \frac{1}{2}\|\beta\|^2 + C\sum\nolimits_{i=1}^{N} y(d(\bar{x}_s, x_i))\|\xi_i\|^2 \\ &\text{Subject to}: t_i\beta \cdot g(x_i) \ge 1 - \xi_i, \quad i = 1, \ldots, N \\ &\qquad \xi_i \ge 0, \quad i = 1, \ldots, N \end{aligned} \tag{12}$$

The first term of the objective function corresponds to the classification margin, while the second term penalizes the model if it misclassifies the training examples. The LELM model is only influenced by training samples that are similar to the testing sample. As a result, the Lagrangian for the optimization problem can be written as:

$$\begin{aligned} L = {} & \frac{1}{2}\|\beta\|^2 + C\sum\nolimits_{i=1}^{N} y(d(\bar{x}_s, x_i))\|\xi_i\|^2 \\ & + \sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{m} \alpha_i(t_i\beta \cdot g(x_i) \ge 1 - \xi_i) + \sum\nolimits_{i=1}^{N} b_i\xi_i \end{aligned} \tag{13}$$

Then, taking derives of L with regard to β, α, ξ_i, we obtain the KKT conditions for the primal problem:

$$\frac{\partial L}{\partial \beta} = 0 \to \beta = \sum_{i=1}^{N} \alpha_i x_i t_i \tag{14}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \to C \cdot y(d(\bar{x}_s, x_i)) - \alpha_i - b_i = 0 \tag{15}$$

$$\frac{\partial L}{\alpha} = 0 \to \beta = t_i\beta \cdot h(x_i) - 1 + \xi_i = 0, \quad i = 1, \ldots, N \tag{16}$$

and satisfy the following conditions:

$$\begin{aligned} &\beta = t_i\beta \cdot h(x_i) - 1 + \xi_i \ge 0, \quad \xi_i \ge 0, b \ge 0 \\ &\alpha_i\{t_i\beta \cdot h(x_i) - 1 + \xi_i\} = 0, \quad b_i\xi_i = 0 \end{aligned}$$

Substituting Eqs. (11–14) into Eq. (10), we obtain the dual form of the LELM optimization problem, which is given as follows:

$$\frac{\partial L}{\alpha} = 0 \to \beta = t_i\beta \cdot h(x_i) - 1 + \xi_i = 0, \quad i = 1, \ldots, N \tag{17}$$

Unlike the optimization problem of ELM, the upper bound for α is modified from C to $y(d(\bar{x}_s, x_i))$. Such modification leads to the following consequence: when $\bar{x}_s$ and x_i are dissimilar, the upper bound will close to zero.

In order to observe the training set rebuilt by LELM algorithm, we select 1000 samples from banana dataset, 800 samples from the training set and 200 samples from the testing set. After LELM algorithm is used to rebuild the training set, only the 100 training samples are retained. Figure 2a shows the data distributions for the two classes, represented as ★ and ▽, respectively. Figure 2b shows that LELM reconstruct the training samples from the original training samples. Figure 2c shows the distribution of the training samples including those removed the training sample in the process that marked by the symbol red ★ and ▽. From Fig. 2, we can known that LELM algorithm not only ignores some noise points, but also keeps the original local structure of the training samples in the process of the classification.

When the testing set is large, the speed will be slow. There is a solution: solve the $L2$ distance of the testing set

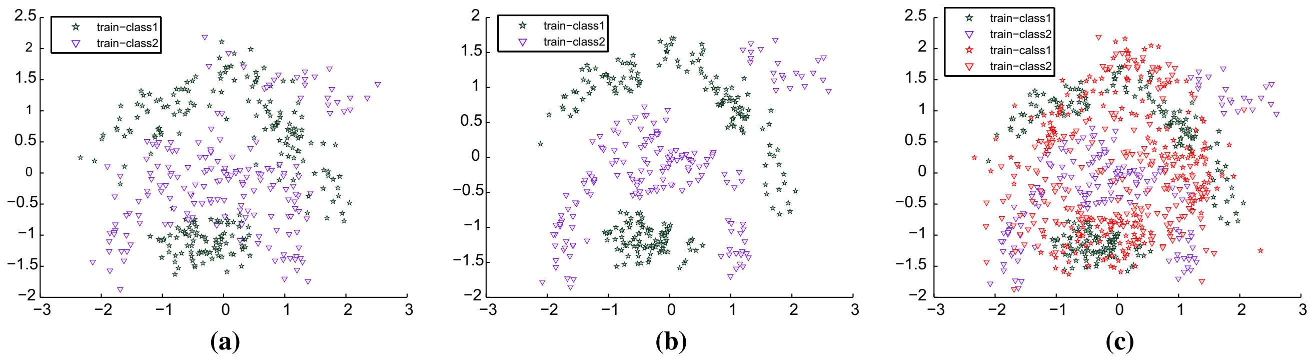

Fig. 2 Contrast of the training set classes distribution. **a** The original training set classes distribution. **b** The training set classes distribution reconstructed by LELM. **c** The original training set classes distribution (including removed samples)

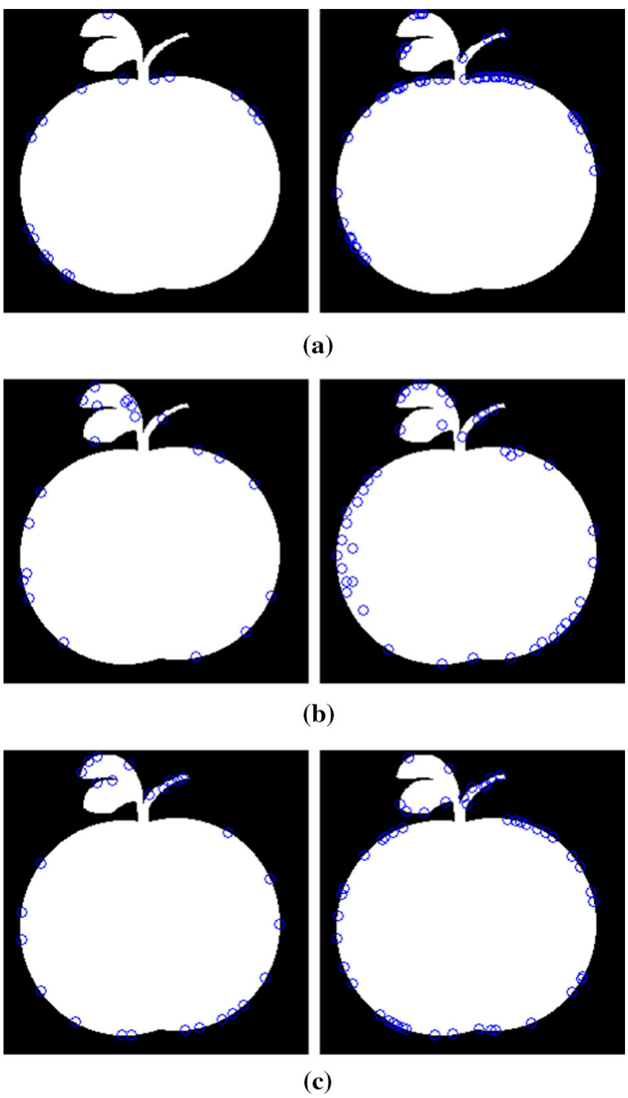

Fig. 3 Comparison between SVM, ELM and LELM in generating feature points, *blue* point as feature points of shape. **a** Use ELM as a decision function to generate feature points. **b** Use SVM as a decision function to generate feature points. **c** Use LELM as a decision function to generate feature points

Table 1 Description of the UCI datasets

Dataset	ALLEx	Atts	Class	testEx
Banana	5300	2	2	1590
Abalone	4177	9	3	1254
Iris	150	4	3	50
Glass	214	9	7	65
Sonar	208	60	2	70
Liver	583	6	2	195
Heart	270	13	2	90
Ionosphere	351	34	2	117
Wine	178	14	3	54

and the training set, so as to pure part of the training samples of the distance.

4 Shape extract

Van et al. [19] solved the image shape extraction problem from a new perspective. This method is based on SVM theory. Firstly, the SVM classifier used SVM to generate the feature points of image shape by solving the binary classification problem. Secondly, select feature points using the gradient. Finally, compute the local feature and apply the different feature to retrieval. The theory is based on SVM; however, as Sect. 2 describes, the SVM classification method exists the following problems:

- SVM is global classification algorithm, so it is hard to keep the sample local structure in the process of classification. And it is difficult to construct a high-dimension global classification model for all training samples.
- The process of classification is an iterative manner to find support vector, and the calculation speed is slow.

- As the optimization equation of SVM has an equality constraint, getting the bigger solution space is impossible.

To solve the above problems, we proposed the LELM shape feature point extraction method and used it for retrieval. The basic steps of the method are as follows:

1. Learn a decision function using the LELM and generate feature points from a given shape,
2. Select feature points by the gradient,
3. Compute local descriptor based on histograms of oriented gradients (HOG) algorithm [28],
4. Retrieval using the dynamic programming [29].

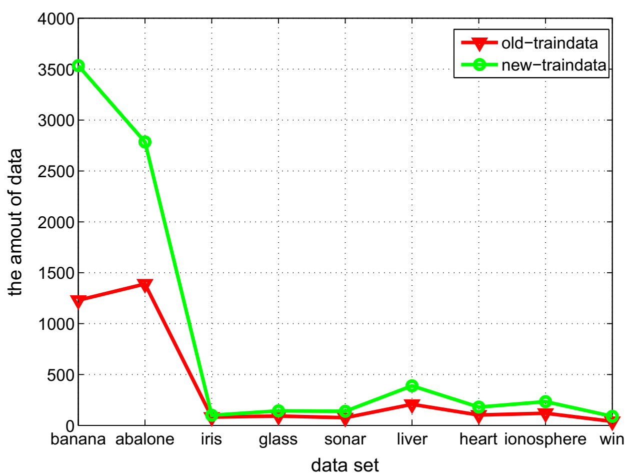

Fig. 4 Comparisons between the amount of the original training samples and the amount of the selected training samples

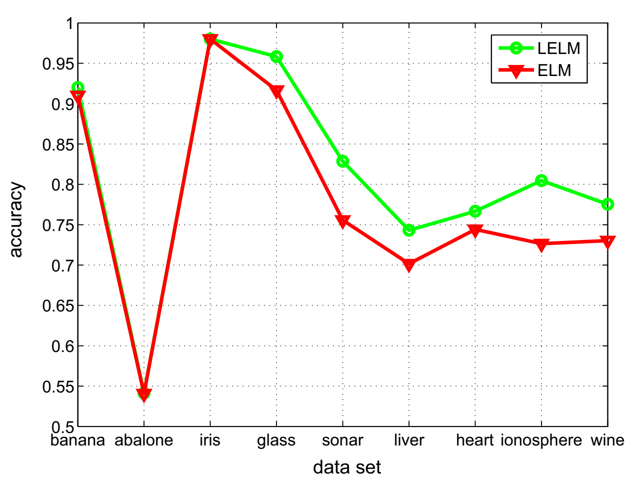

Fig. 5 Classification accuracy for ELM and LELM on the UCI dataset

4.1 Generate feature points

SVS uses kernel SVM classification model to extract the image shape feature points [19]. And then, it has been analyzed in theory and verified by simulations over extensive of applications that SVM may provide suboptimal solutions to ELMs if the same kernels are used. LELM classification model has a bigger solution space than SVM. At the same time, it takes into account local construct of samples and avoids the global variance and bias imbalance for high-dimensional data classification. Based on above points, in the paper, we use the kernel-LELM to extract the feature point.

Image shape extraction problem is converted into a binary classification problem, and the steps are as follows:

1. Get class samples: the pixel within the graphic is marked class 1, the graphic edge and external pixel is marked as class 0, and get a set (x_i, y_i).
2. Train classification model: we use LELM as a classifier, combined with cross-validation method to get the classification boundaries. The radial basic function (RBF) kernel is used with LELM to make our representation rotational and translational invariant.
3. Get feature points: through calculation, we obtain β, the sample with the smaller β value is the feature point; in other words, the smaller β corresponds to the sample that is the feature points of the shape.

4.2 Select feature points

In order to reduce the amount of image retrieval calculation, selecting better feature points is needed. The author of [19] analyzed the varieties of a kind of selection methods based on the gradient method, the curvature method, the information entropy method and the random select method. According to the author's experimental results, the method based on the gradient method is better than other methods for image retrieval. And it is concluded that the gradient orientation is stable, while gradient magnitude differs only with a constant factor. So we select feature points of bigger gradient amplitude. Figure 3 shows the feature points using SVM, ELM and LELM. The left part of Fig. 3 is the top 30

Fig. 6 Typical shape images from the MPEG7 shape dataset

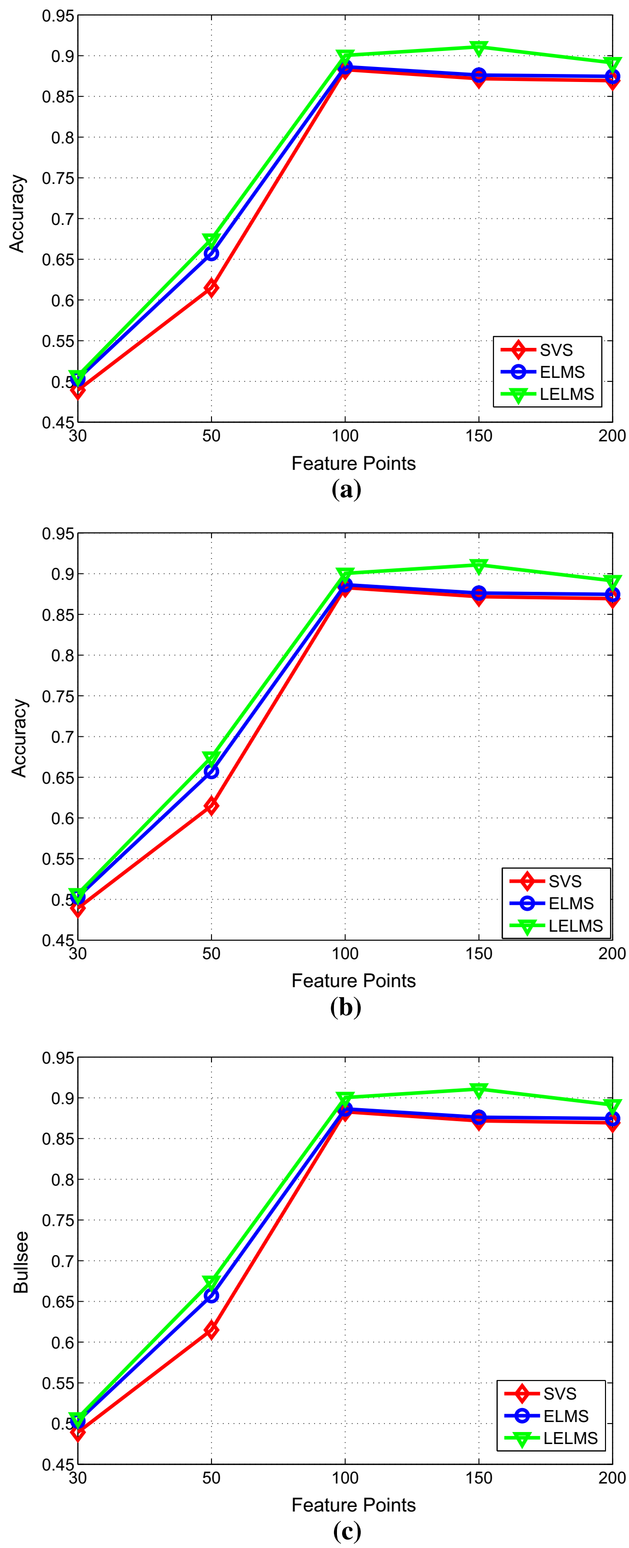

Fig. 7 Comparison of the retrieval performance of three methods, when the number of feature points is different. **a** The retrieval accuracy of top 10. **b** The retrieval accuracy of top 20. **c** Bullseye test

Table 2 Bullseye test results on MPEG7 shape dataset

Method	DISC-DP	SVS-DP	ELMS-DP	LELMS-DP
Accuracy	77.21	87.18	87.62	91.08

gradient amplitude points, and the right part is the top 50 gradient amplitude points. In Fig. 3c, it can be intuitively shown that LELM extraction feature point method is better than SVM and ELM. When the shape of the image edge is changeable, it is needed to maintain the local structure of the edge data. In this situation, taking advantage of LELM can keep the local structure of the data. Figure 3 shows that with LELM method only fewer feature points can better express the image shape.

4.3 Compute local descriptors

In this paper, the HOG [28] algorithm is used to compute feature vector that surrounds each feature point around. HOG parameter settings: bins is 8, block size is 2×2, the feature vector dimensions is 32.

4.4 Match feature point

Dynamic programming (DP) method [29] is used to match the feature points between the detecting image and the target image: the feature point sequences $a_1, a_2, \ldots, a_p$ for the detecting image A and the feature point sequences $b_1, b_2, \ldots, b_q$ for the target image B, assuming $p \geq q$. The matching $\pi(v)$ from A to B is a mapping from 1, 2,..., p to 0, 1, 2,...,q, where a_v is matched to $b_{\pi(v)}$ if $\pi(v) \neq 0$, $\pi(v)$ should minimize the match cost function which is defined as:

$$C(\pi) = \sum_{v=1}^{p} c(v, \pi(v))$$

where the distance is computed using χ^2 statistic [31] :

$$c(v, \pi(v)) = \sum_{k=1}^{32} \frac{[A_{\pi(v)}(k) - B_v(k)]^2}{A_{\pi(v)}(k) + B_v(k)}$$

Note that the mapping $\pi(v)$ is neither one-to-one nor overlapping, but keeps the order of the descriptors.

5 Experiment

This paper implements LELM algorithm, which is used to generate feature points of image shape. First, the UCI dataset is used to verify the validate of LELM. Then, the shape feature extraction method is tested in MPEG7 shape

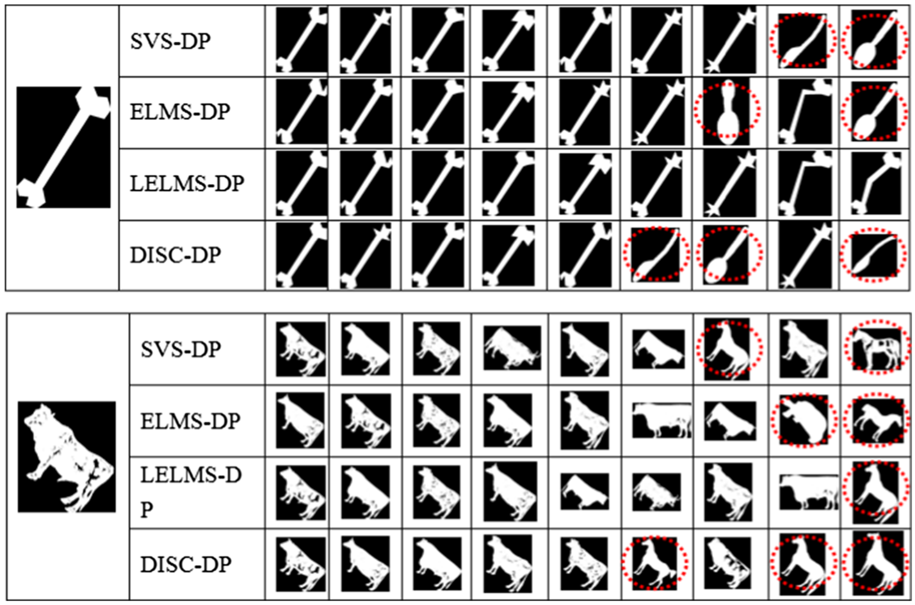

Fig. 8 Comparison of the retrieval results on MPEG7 shape dataset. *Red circles* show incorrect matches

Table 3 Bullseye test results for random sheltered image on the MPEG7 shape dataset

Method	DISC-DP	SVS-DP	ELMS-DP	LELMS-DP
Accuracy	50.68	91.27	91.73	92.82

database and Kimia database. DP algorithm is used to match feature points in the image retrieval process.

5.1 UCI classification dataset

In order to test the performance of LELM algorithms, we selected 9 datasets from the UCI dataset. Table 1 describes the data used in the experiment in detail.

LELM reconstructs the training set, so we show the comparison between the original training set and the selected training set in Fig. 4 where the red line stands for the amount of data selected by LELM. We observe that the proposed LELM reduces the size of the training set and reduces computational complexity. Figure 5 shows the classification accuracy comparison between ELM and LELM. It is observed that LELM outperforms ELM in accuracy. This verifies that LELM is simple and effective for classification.

5.2 MPEG7 shape dataset

MPEG7 dataset has 70 classes and 20 shapes for each class, a total of 1400 images. Figure 6 shows all typical images.

We calculate the retrieval effectiveness in occupations that the feature points are 30, 50, 100, 150, 200. Each category of the feature point contains three types: the retrieval accuracy of top 10, the retrieval accuracy of top 20 and the calculation method of the Bullseye. The Bullseye is as follows: firstly, we calculate similarities between the target image and the entire 1400 images in dataset. Secondly, based on the similarity sort, we get 40 images with the maximum similarity. Finally, we compute the ratio of the correct number of 40 images in 20 correct images which are the total correct images in MPEG7 database.

Figure 7 shows the comparison between SVS-DP, ELM-DP and LELM-DP retrieval performance from different feature point: the first image is the retrieval accuracy of top 10, the second image is the retrieval accuracy of top 20, the last image is Bullseye. Results indicate that with the increase in feature points, retrieval performance improved. All of methods have the best retrieval performance when the number of feature points is 150, and LELM-DP is better than others.

We reproduce IDSC-DP that is a popular image retrieval algorithm in this experiment. Parameters are set as follows: the inner-distance bins is 8, the number of inner-angle bins is 12, the number of relative orientation bins is 8, the penalty is 0.6. Bullseye test results on IDSC-DP, SVS-DP, ELM-DP, LELM-DP are summarized in Table 2. The number of the feature points of three algorithms (SVS-DP, ELM-DP, LELM-DP) is 150. As can be seen from the table, LELM-DP outperforms other three algorithms. The performance of IDSC-DP is 77.21 %, and the severe change in the shape makes the inner distances failure. SVS-DP not work well enough, because the classification performance of SVM is worser than LELM.

Figure 8 shows the retrieval results of top 10 image using four algorithms (IDSC-DP, SVS-DP, ELM-DP, LELM-DP). It can be seen from the figure that LELM-DP

Fig. 9 Comparison of the retrieval results for the sheltered image on MPEG7 shape dataset. *Red circles* show incorrect matches

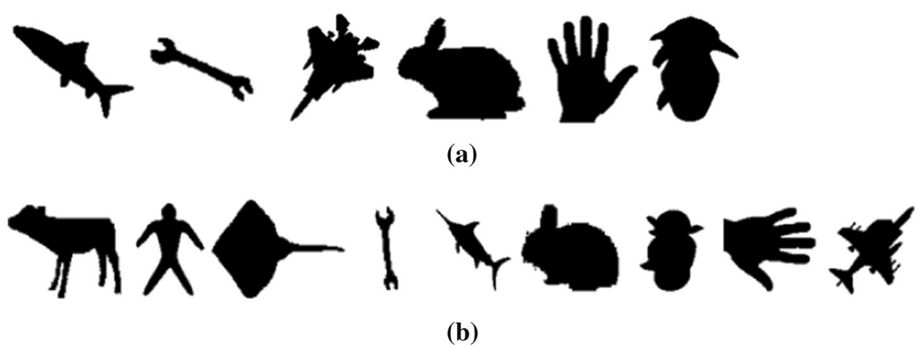

Fig. 10 **a** Typical shape images from the Kimia1 dataset. **b** Typical shape images from the Kimia2 dataset

retrieval results are better than other algorithms. Red circles show error matches, and IDSC-DP is indistinguishable from similar shapes.

Table 3 compares the results in the case of a sheltered between IDSC-DP, SVS-DP, ELM-DP and LELM-DP, and the number of the feature points is 150 in SVS-DP, ELM-DP, LELM-DP. As shown in Table 3, DISC-DP algorithm is not enough stable, and for a block of images retrieval results are poor, but the effect LELM-DP retrieval algorithm is better than other three algorithms. Figure 9 shows the comparison of retrieval results between SVS-DP, ELM-DP, LELM-DP and DISC-DP algorithms in descending order of the matching scores. LELM-DP algorithm has better stability than other algorithms for sheltered images.

5.3 The Kimia dataset

Algorithm is tested on two Kimia shape datasets [30, 31]. The first image shape dataset contains 6 classes of data that contain 25 images. Figure 10a gives all typical images. The Kimia2 dataset contains 9 classes, a total of 99 images, Fig. 10b shows that all typical images.

Table 4 Retrieval result on Kimia1 dataset

Method	Top 1	Top 2	Top 3
IDSC-DP	25/25	24/25	25/25
SVS-DP	25/25	24/25	25/25
ELMS-DP	25/25	24/25	24/25
LELMS-DP	25/25	25/25	25/25

Retrieval results are summarized in Table 4. The numbers of Top1, Top2 and Top3 which images match with the retrieval target have been counted. It shows that LELM-DP slightly outperforms the other three algorithms.

Table 5 summarizes the retrieval results of Kimia2 dataset, the number of top 1 to top 10 closest matches (the best possible result for each of them is 99). It can be seen

Table 5 Retrieval result on Kimia2 dataset

Method	1	3	5	6	7	8	9	10
IDSC-DP	99/99	99/99	98/99	97/99	97/99	94/99	98/99	79/99
SVS-DP	99/99	99/99	99/99	98/99	97/99	97/99	96/99	87/99
ELMS-DP	99/99	99/99	99/99	97/99	97/99	98/99	94/99	88/99
LELMS-DP	99/99	99/99	99/99	98/99	97/99	98/99	96/99	90/99

from Table 5, LELM-DP retrieval performance is superior to other algorithms.

6 Conclusion

This paper proposed a local extreme learning machine algorithm which sufficiently considers the local structure of training samples and balances between the variance and bias of high-dimensional data. This algorithm is more effective than traditional algorithm in solving the problem of extracting the feature points of image. The 3D object image retrieval has becoming a hot in image processing, with the 3D images being widely used. In our next works we hope to make an improvement in this aspect.

Acknowledgments This research is supported by National Natural Science Foundation of China Nos. 61173163, 51105052, 61370200.

References

1. Gonzalez Rafael C, Woods Richard E (2010) Digital image processing, 3rd edn. Electronic Industry Press, Beijing
2. ISO/IEC JTC1/SC29/WG11N6828, MPEG-7 Overview (version 10)
3. Zhang D, Lu G (2004) Review of shape representation and description techniques. Pattern Recognit 37(1):1–19
4. Mokhtarian F (1995) Silhouette-based isolated object recognition through curvature scale space. IEEE Trans Pattern Anal Mach Intell 17(5):539–544
5. Mokhtarian F, Bober M (2011) Curvature scale space representation: theory, applications, and MPEG-7 standardization. Springer Publishing Company, Incorporated
6. Abbasi S, Mokhtarian F, Kittler J (1999) Curvature scale space image in shape similarity retrieval. Multimed Syst 7(6):467–476
7. Lien JM, Amato NM (2008) Approximate convex decomposition of polyhedra and its applications. Comput Aided Geom Des 25(7):503–522
8. Liu H, Liu W, Latecki LJ (2010) Convex shape decomposition[C]//Computer Vision and Pattern Recognition (CVPR), In: 2010 IEEE conference on IEEE, pp 97–104
9. Adamek T, O'Connor NE (2004) A multiscale representation method for nonrigid shapes with a single closed contour. IEEE Trans Circuits Syst Video Technol 14(5):742–753
10. Ling H, Jacobs DW (2007) Shape classification using the inner-distance. IEEE Trans Pattern Anal Mach Intell 29(2):286–299
11. Hua J, Lai Z, Dong M, Gu X, Qin H (2008) Geodesic distance-weighted shape vector image diffusion. IEEE Trans Vis Comput Graph 14(6):1643–1650
12. Gorelick L, Galun M, Sharon E et al (2006) Shape representation and classification using the poisson equation. IEEE Trans Pattern Anal Mach Intell 28(12):1991–2005
13. Cohen-Steiner D, Alliez P, Desbrun M (2004, August) Variational shape approximation. In ACM Transactions on Graphics (TOG). ACM, vol 23, No. 3, pp 905–914
14. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 79(1):12–49
15. Malladi R, Sethian JA, Vemuri BC (1995) Shape modeling with front propagation: a level set approach. IEEE Trans Pattern Anal Mach Intell 17(2):158–175
16. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 24(4):509–522
17. Mori G, Belongie S, Malik J (2005) Efficient shape matching using shape contexts. IEEE Trans Pattern Anal Mach Intell 27(11):1832–1837
18. Van Nguyen H, Porikli F (2013) Support vector shape: a classifier-based shape representation. IEEE Trans Pattern Anal Mach Intell 35(4):970–982
19. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
20. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
21. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
22. Zong W, Huang GB (2011) Face recognition based on extreme learning machine. Neurocomputing 74(16):2541–2551
23. Li MB, Huang GB, Saratchandran P, Sundararajan N (2005) Fully complex extreme learning machine. Neurocomputing 68:306–314
24. Huang GB, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74(1):155–163
25. Bai Z, Huang GB, Wang D et al (2014) Sparse extreme learning machine for classification. IEEE Trans Cybern 44(10):1858–1870
26. 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
27. Huang GB (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cogn Comput 6(3):376–390
28. Dalal N, Triggs B (2005, June) Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on IEEE, vol 1, pp. 886–893
29. Cover TM (1968) Estimation by the nearest neighbor rule. IEEE Trans Inf Theory 14(1):50–55
30. Sharvit D, Chan J, Tek H, Kimia BB (1998, June) Symmetry-based indexing of image databases. In Content-Based Access of Image and Video Libraries, 1998. In: Proceedings. IEEE workshop on IEEE, pp 56–62
31. Sebastian TB, Klein PN, Kimia BB (2004) Recognition of shapes by editing their shock graphs. IEEE Trans Pattern Anal Mach Intell 26(5):550–571