

An improved radial basis function neural network for object image retrieval

Gholam Ali Montazer^{a,b,*}, Davar Giveki^b

^a Information Technology Engineering Department, School of Engineering Tarbiat Modares University, P.O. Box 14115-179, Tehran, Iran

^b Iranian Research Institute for Information Science and Technology (IranDoc), Tehran, Iran

ARTICLE INFO

Article history:

Received 7 November 2014

Received in revised form

23 May 2015

Accepted 29 May 2015

Communicated by K. Li

Available online 10 June 2015

Keywords:

Radial basis function neural networks (RBFNNs)

Improved Particle Swarm Optimization

Optimum Steepest Descent (OSD)

Object Image Retrieval

ABSTRACT

Radial Basis Function Neural Networks (RBFNNs) have been widely used for classification and function approximation tasks. Hence, it is worthy to try improving and developing new learning algorithms for RBFNNs in order to get better results. This paper presents a new learning method for RBFNNs. An improved algorithm for center adjustment of RBFNNs and a novel algorithm for width determination have been proposed to optimize the efficiency of the Optimum Steepest Descent (OSD) algorithm. To initialize the radial basis function units more accurately, a modified approach based on Particle Swarm Optimization (PSO) is presented. The obtained results show fast convergence speed, better and same network response in fewer train data which states the generalization power of the improved neural network. The Improved PSO–OSD and Three-phased PSO–OSD algorithms have been tested on five benchmark problems and the results have been compared. Finally, using the improved radial basis function neural network we propose a new method for object image retrieval. The images to be retrieved are object images that can be divided into foreground and background. Experimental results show that the proposed method is really promising and achieves high performance.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Radial basis function neural networks approaches, developed by Broomhead and Lowe in 1988 [1], are feed-forward networks which are trained by a supervised algorithm. They have been broadly used in classification and interpolation regression tasks [2,3]. Comparing to other neural networks, the RBFNNs are faster in their training phase and provide a better approximation due to their simpler network architecture. Since training is a very important issue of the RBFNNs, enhancing their learning algorithms have been addressed by previous works [4–6]. The RBFNNs are three-layer structures including an input layer, a hidden layer, and an output layer. The input layer contains n inputs which connects the input space to the environments. The hidden layer consists of k Radial Basis Function (RBF) units. This layer transforms the input space to the hidden space whose dimensionality is higher than the input layer. Each hidden unit locally estimates the similarity between an input pattern and its connection weights or centers.

The output layer, consisting of m linear units, produces output to the input pattern. These networks carry out the mapping $f: \mathcal{R}^n \rightarrow \mathcal{R}^m$ such that

$$y_s(P) = \sum_{j=1}^k w_{js} \varphi \left(\frac{\|P - C_j\|}{\sigma_j} \right) \quad \text{for } 1 \leq s \leq m, \quad (1)$$

where y_s is the s th network output, P is an input pattern, and w_{js} is the weight of the link between j th hidden neuron and s th output neuron. Furthermore, C_j and σ_j are the center and the width of the j th RBF unit in the hidden layer, respectively. In Eq. (1), the term φ refers to an activation function such as Gaussian function, defined as the following [7]:

$$\varphi(r) = e^{-r^2}, \quad (2)$$

where r is the variable of radial basis function φ . RBFNNs have been trained by different learning algorithms in the previous researches (we will name a few of them in the following), where all of them comprised two main steps. In the first step, the parameters of the hidden layer, the centers C_j and the widths σ_j of the RBF units, are computed. The algorithms which have been previously developed for this step are regularization approach [8], Expectation–Maximization (EM) [9], randomly selection of the fixed centers and the supervised selection of the centers [10], orthogonal least squares [11], self-organized selection of centers using K -means clustering,

* Corresponding author at: Information Technology Engineering Department, School of Engineering Tarbiat Modares University, P.O. Box 14115-179, Tehran, Iran. Tel.: +98 9123230540.

E-mail addresses: montazer@modares.ac.ir (G.A. Montazer), Giveki@students.irandoc.ac.ir (D. Giveki).

¹ Visiting Lecturer.

the self-organizing feature map clustering [12] and regression tree [13].

The second step is estimating the connection weights. The previously introduced algorithms for connection weight estimation are the Least-Mean-Square (LMS) [6], the Steepest Decent (SD) [14], the pseudo-inverse (minimum-norm) [15], and the Quick Propagation (QP) [16]. In the previous work [16], the Optimum Steepest Decent (OSD) method has been introduced for calculating the connection weights which uses an optimum learning rate in each epoch of the training process.

In spite of its fast learning process and high performance, it suffers from random selection of the centers and widths of the RBF units, which decreases the efficiency of the proposed RBFNN.

As a follow-up work, authors in [17] propose a three-phase learning algorithm to improve the performance of the OSD. This method uses K -means and p -nearest neighbor algorithms to determine the centers and the widths of RBF units, respectively, which results in a greater precision in initializing RBF unit centers and widths. This algorithm guarantees reaching the global minimum in the weight space, however, the sensitivity of K -means to the center initialization can lead the algorithm to get stuck in a local minimum which results in a suboptimal solution.

Authors in [5] propose a new learning method, called Three-Phased PSO–OSD, in which K -means clustering is replaced with a new clustering method using PSO algorithm. This makes the algorithm rather stable against the center initialization. Moreover, it has been shown that the training process using PSO is repeatable.

Although this approach outperforms the existing learning methods used in RBFNNs, it is slow to some degree due to the nature of PSO clustering. In addition, using p -nearest neighbor algorithm for computing the widths of RBF units results in a loss of information about the spatial distribution of the training dataset; and as a consequence, the computed widths do not make a major contribution in the classification performance.

In this paper, our main motivations are to deal with the shortcomings of the Three-Phased PSO–OSD [5] and to introduce an efficient object image retrieval method.

Thus, our main contributions are proposing new center adjustment and width determination methods. A new method for center adjustment makes an improvement to the PSO clustering in order to speed up its convergence and also to increase its classification performance. Proposed method for width determination improves the classification performance of the Three-Phased PSO–OSD even if we use the conventional PSO clustering proposed in [5].

Proposing a new (object) image retrieval method using Discrete Wavelet Transform (DWT) as well as a strategy for the object background elimination are other contributions of this paper. The main reason behind using wavelet transform in a feature extraction task is that it is computationally cheap and the resulted feature vectors are from a low dimensionality while they are discriminant enough. Moreover, it has been successfully applied in Content Based Image Retrieval (CBIR) and image classification scenarios with high performance [18–22].

Experimental results show that our improvements to the Three-phased PSO–OSD enhance the RBFNN's classification accuracy to a high degree. Furthermore, it has been experimentally shown that using DWT features, satisfactory results are achieved especially when the method is applied in CIELAB color space and the features are extracted from the image foreground.

The rest of the paper is organized as follows. In Section 2 a modified PSO and a novel way of computing the widths of RBFNN are described. In Section 3 we explain a new method for content based object image retrieval using the improved PSO–OSD RBFNN and wavelet transform (WT). Experimental results of the improved PSO–OSD on benchmark datasets and experimental results of applying our proposed method on Caltech 101 dataset are

discussed in Section 4. The concluding discussion is presented in Section 5.

2. Improved PSO–OSD radial basis function neural network

The recent method developed for RBFNN, the PSO–OSD [5], introduced a new PSO clustering for initializing centers of the hidden layer units of the Gaussian functions of RBFNN. Additionally, it used OSD algorithm to train the proposed RBFNN. The authors in [16,23,24] introduced a new learning method to improve the RBFNN center designing and their learning rate parameters. They showed a significant increase in the classification accuracy and the convergence speed of the new neural network in real world problems. Inspired by the previous works, in this paper, we further improve the PSO–OSD in terms of computing the centers and the widths of the radial basis functions.

2.1. Computing the centers of the RBFNN using improved PSO

PSO is a stochastic population-based optimization algorithm, where the members of the population are called “particles”. In this algorithm, each particle flies in a multi-dimensional search space, where its velocity is constantly updated by the particle's own experience and the experience of the neighboring particles (the experience of the whole swarm).

In the proposed PSO of [5], the velocity and position updating rules are given by

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (pbest_{id}^k - x_{id}^k) + c_2 r_2 (gbest_d^k - x_{id}^k) \quad (3)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, \quad i = 1, 2, \dots, n \quad (4)$$

$$\omega = \omega_{\max} - \frac{k}{k_{\max}} (\omega_{\max} - \omega_{\min}) \quad (5)$$

where the current position of the particle i in the k th iteration is x_{id}^k and v_{id}^k is the current velocity of the particle which is used to determine the new velocity v_{id}^{k+1} . The c_1 and c_2 are acceleration coefficients. The r_1 and r_2 are two independent random numbers uniformly distributed in the range of [0,1]. In addition, $v_i \in [-v_{\max}, v_{\max}]$, where $v_{\max}$ is a problem-dependent constant defined to clamp the excessive roaming of particles. The $pbest_{id}^k$ is the best previous position along the d th dimension of the particle i in the iteration k (memorized by every particle); $gbest_d^k$ is the best previous position among all the particles along the d th dimension in the iteration k (memorized in a common repository). The $\omega_{\max}$ and $\omega_{\min}$ are the maximum and the minimum of ω , respectively. The $k_{\max}$ is the maximum number of iterations.

In spite of its high optimization ability, PSO can get trapped in a local optimum which slows down the convergence speed. In this paper, we propose an adaptive strategy to the conventional PSO as follows:

$$v_{id}^{k+1} = \omega_i^k v_{id}^k + c_1 r_1 (pbest_{id}^k - x_{id}^k) + c_2 r_2 (gbest_d^k - x_{id}^k) \quad (6)$$

$$x_{id}^{k+1} = x_{id}^k + \mu_i^k v_{id}^{k+1}, \quad i = 1, 2, \dots, n \quad (7)$$

The adaptive strategy is a method to dynamically adjust the inertia weight factor ω and the new velocity v_{id}^{k+1} by introducing the coefficient μ .

The inertia weight ω has a great influence on the optimal performance. Empirical studies of PSO with inertia weight have shown that a relatively large ω has more global search ability while a relatively small ω results in a faster convergence. Although in Eq. (3), ω is adaptive, it is updated using the linear updating strategy of Eq. (5). As a result, ω is just relevant to the current iteration and maximum number of iterations (k and $k_{\max}$) and cannot adapt to

the characteristics of complexity and high nonlinearity. If the problem is extremely complex, the global search ability is insufficient in the later iteration. Therefore, in order to overcome the above defects, an improved method for updating ω is proposed.

Generally, we expect particles to have strong global search ability in the early evolutionary search while strong local search ability in the late evolutionary search. This makes particles find the global optimal solution.

In order to get better search performance, the dynamic adjustment strategy for ω and μ is proposed as follows:

$$\omega_i^k = k_1 h_i^k + k_2 b_i^k + \omega_0 \quad (8)$$

$$h_i^k = |(\max\{F_{id}^k, F_{id}^{k-1}\} - \min\{F_{id}^k, F_{id}^{k-1}\}) / f_1| \quad (9)$$

$$b_i^k = 1/n * \sum_{i=1}^n (F_i^k - F^{avg}) / f_2 \quad (10)$$

$$\mu_i^k = \begin{cases} (v_{\min} / v_i^k) \cdot e^{(k/k_{\max})^2} & \text{if } v_i^k > v_{\max} \\ e^{(k/k_{\max})^2} & \text{if } v_{\min} < v_i^k < v_{\max} \\ v_{\min} & \text{if } v_i^k < v_{\min} \end{cases} \quad (11)$$

where $\omega_0 \in (0, 1]$ is the inertia factor which manipulates the impact of the previous velocity history on the current velocity (in most cases is set to 1). In Eq. (8), coefficients k_1 and k_2 are typically selected experimentally within the range of $[0, 1]$. In Eq. (11), the parameter μ adaptively adjust the value of v^{k+1} by considering the value of v^k . h_i^k is the speed of evolution, b_i^k is the average fitness variance of the particle swarms, F_{id}^k is the fitness value of $pbest_{id}^k$ namely $F(pbest_{id}^k)$, F_{id}^{k-1} is the fitness value of $pbest_{id}^{k-1}$ namely $F(pbest_{id}^{k-1})$, f_1 is the normalization function, $f_1 = \max\{\Delta F_1, \Delta F_2, \dots, \Delta F_n\}$, $\Delta F_i = |F_{id}^k - F_{id}^{k-1}|$, n is the size of the particle swarms, F_i^k is the current fitness of the i th particle, F^{avg} is the mean fitness of all particles in the swarm at the k th iteration, f_2 is the

normalization function, $f_2 = \max\{|F_1^k - F^{avg}|, |F_2^k - F^{avg}|, \dots, |F_n^k - F^{avg}|\}$. The dynamic adjustment helps PSO not only to avoid the local optimums, but also to enhance the population diversity, which in turn improves the quality of solutions.

In order to compute the RBFNN centers using the improved PSO algorithms such as the method proposed in [5], suppose that a single particle represents a set of k cluster centroid vectors $X = (M_1, M_2, \dots, M_k)$, where $M_j = (s_{j1}, \dots, s_{jl}, \dots, s_{jf})$ refers to the j th cluster centroid vector of a particle. The M_j has f columns representing the number of features for each pattern of the dataset. Each swarm contains a number of data clustering solutions. The Euclidean distance between each feature of the input pattern and the corresponding cluster centroids is measured by

$$d(M_{jl}, P_{rl}) = \sqrt{\sum_{l=1}^f (s_{jl} - t_{rl})^2} \quad \text{for } 1 \leq j \leq k, 1 \leq r \leq n, 1 \leq l \leq f. \quad (12)$$

After computing all distances for each particle, feature l of the pattern r is compared with the corresponding feature of the cluster j , and then assigns 1 to $Z_{jrl} = 1$ when the Euclidean distance for each feature l of the pattern r is minimum:

$$Z_{jrl} = \begin{cases} 1 & d(M_{jl}, P_{rl}) \text{ is min} \\ 0 & \text{elsewhere} \end{cases} \quad (13)$$

In a next step, the mean of the data N_{jl} is computed for each particle according to

$$N_{jl} = \frac{\sum_{r=1}^n t_{rl} \times Z_{jrl}}{\sum_{r=1}^n Z_{jrl}} \quad \text{for } 1 \leq j \leq k, 1 \leq l \leq f \quad (14)$$

Moreover, for each feature l of the cluster j , the Euclidean distances between mean of data N_{jl} and the centroid S_{jl} are computed by

$$d(N_{jl}, S_{jl}) = \sqrt{(N_{jl} - S_{jl})^2} \quad \text{for } 1 \leq j \leq k, 1 \leq l \leq f \quad (15)$$

Now, the fitness function for each cluster is obtained by summing

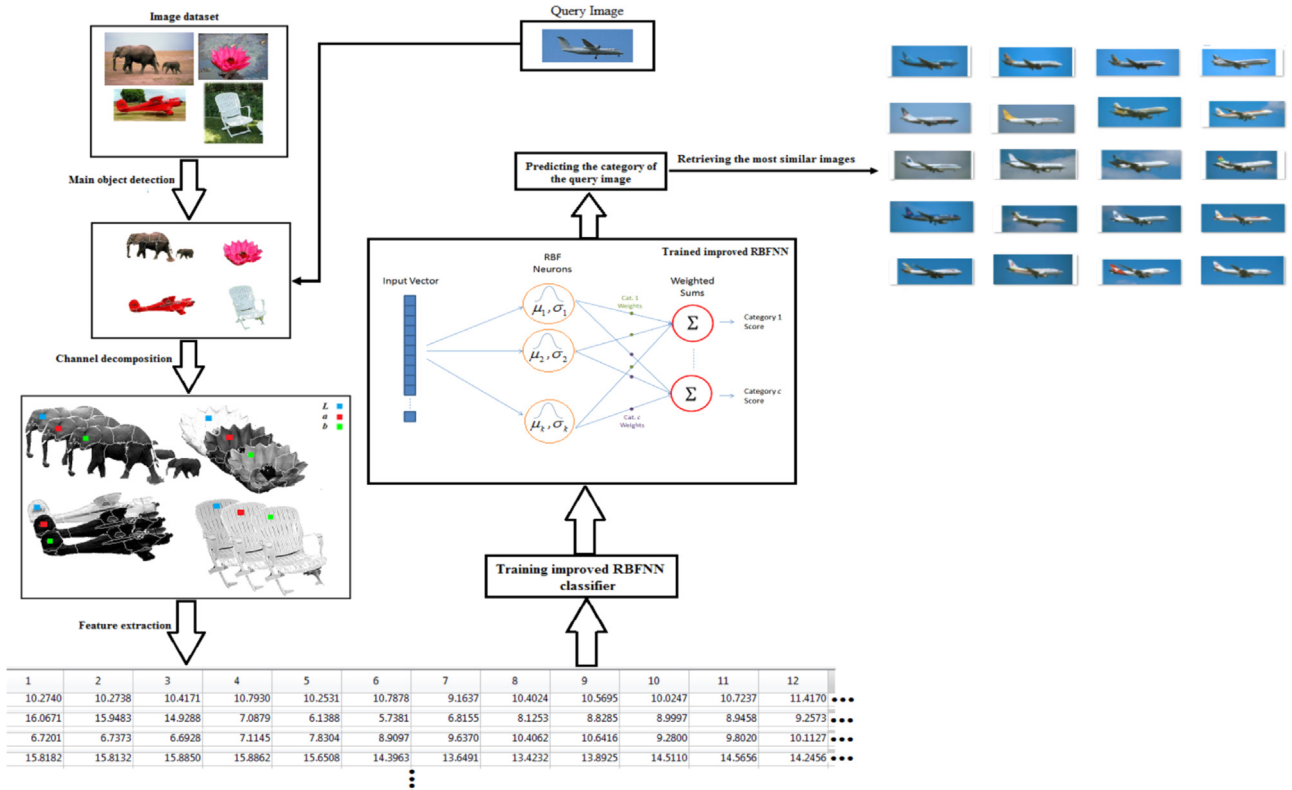


Fig. 1. The structure of the proposed model for object image retrieval.

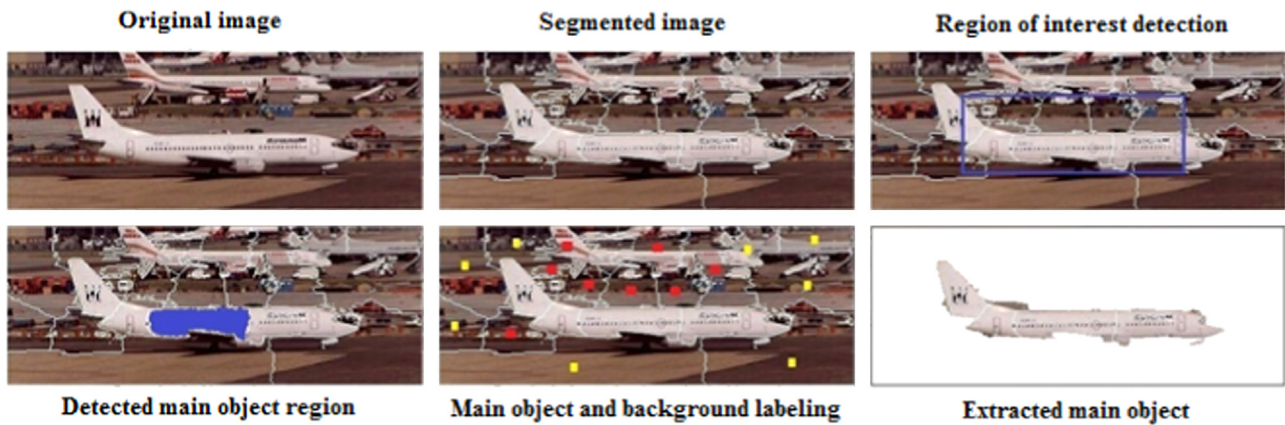


Fig. 2. The steps of the proposed main object detection algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the calculated distances as follows:

$$F(M_j) = \sum_{l=1}^f d(N_{jl}, S_{jl}) \quad \text{for } 1 \leq j \leq k \quad (16)$$

The proposed method has been shown in Algorithm 1. Using this algorithm, we adjust RBF unit centers with *gbest* by iterating for a $k_{\max}$ number of iterations.

Algorithm 1. The pseudocode of the proposed PSO clustering for RBF unit center.

```

1: for each Particle [i] do
2:   Initialize Position vector X [i] in the range of maximum and
   minimum of dataset patterns
3:   Initialize Velocity vector V [i] in the range of [-a,a](a=max
   (data)-min(data))
4:   Put initial Particle[i] into pbestid [i]
5: for end
6: while maximum iteration is not attained do
7:   for each Particle [i] do
8:     for each Cluster [j] do
9:       Compute the Fitness Function using Eqs. (14)–(16)
10:    end for
11:  end for
12:  if run number is greater than 1 then
13:    for each Particle [i] do
14:      for each Cluster [j] do
15:        if Fitness Function of Particle [i]'s Cluster [j] is better
        than Fitness Function of pbestid [i]'s Cluster [j] then
16:          Put Cluster [j] of Particle [i] into Cluster [j] of pbestid
          [i]
17:        end if
18:      end for
19:    end for
20:  end if
21:  for each Cluster [j] do
22:    for each Particle [i] do
23:      Put the best of pbest in terms of Fitness Function into
      gbest
24:    end for
25:  end for
26:  Compute inertia weight  $\omega$  using (8)
27:  Compute using (11)
28:  for each Particle [i] do
29:    Update Velocity vector V [i] using (6)
31:    Update Position vector X [i] using (6)
32:  end for

```

33: Having processes algorithm, RBF unit centers are adjusted with *gbest*

34: **end while**

2.2. Computing the widths of the RBFNN

Although the setting of the basis function centers has been highly addressed by the previous works on RBFNN learning [25–27], the learning of the basis function widths has not been much studied. The existing previous works discussed the effect of widths of radial basis functions on performances of classification and function approximation [4,28,29].

Being aware of the high importance of the spatial distribution of the training dataset and the nonlinearity of the function, whose approximation is desired, we take into account them for the classification problem. Thus, the Euclidean distances between center nodes and the second derivative of the approximated function are used to measure these two factors.

Since the width of the center nodes within highly nonlinear areas should be smaller than those of the center nodes in flat areas, we compute the widths of the RBFNN in our experiments according to Algorithm 2.

Algorithm 2. The algorithm of the proposed width adjustment.

```

1: Compute the centers of the radial basis functions using our
   improved PSO clustering
2: Compute mean of squared distances between the centre of
   cluster j and p-nearest neighbors
3: Compute coefficient factor: coeff =  $d_{\max}/N$ , where N is the
   number of hidden units and  $\pm bd_{\max}$  is the maximum
   distance between those centers.
4: Find the maximum distance from each center and normalize
   the distance vector
5: Multiply the distance vector obtained from step 4 by the
   coefficient factor
6: Sum the vector obtained from step 5 with the vector obtained
   from step 2 as the widths of the improved PSO–OSD RBFNN

```

In the case of having a function approximation problem, the widths can be computed using Eq. (17) as follows:

$$\sigma_i = \frac{d_{\max} r_i}{\sqrt{N} \cdot \bar{r}} \cdot \left[\frac{1}{1 + |f'(c_i)|} \right]^{1/4} \quad (17)$$



Fig. 3. Layout of individual bands at second level of DWT decomposition.

Table 1

Description of the five benchmark problems.

Data set	No. of features	No. of classes	No. of patterns
Iris	4	3	150
Wine	13	3	178
Abalone	8	3	4177
Ionosphere	34	2	351
Glass	9	6	214

Table 2

Description and values of the parameters used in the PSO–OSD.

Parameters	Description	Considered value
ω_{max}	Maximum inertia weight	1.1
ω_{min}	Minimum inertia weight	0.7
C1	Local search coefficient	1.5
C2	Social search coefficient	1.5
Population size	Number of particles	15
k_{max}	Number of iterations	400
Number of runs	–	50

For center node c_i , the average distance between this node and its p -nearest neighbor nodes is used to measure the spatial distribution feature at this center node, which is defined as the following:

$$r_i = \frac{1}{p} \left(\sum_{j=1}^p \|c_j - c_i\|^2 \right)^{1/2} \quad (18)$$

where c_j are the p -nearest neighboring nodes of c_i . The r_i is the reference dense distance at c_i and the $\bar{r}$ is the average of reference dense distances of all the center nodes and it is computed as follows:

$$\bar{r} = \frac{1}{N} \left(\sum_{i=1}^N r_i \right) \quad (19)$$

$f''(c_i)$ is the second derivative of function f and in point c_i and can be computed using the central finite difference method. As the second derivative is used to measure the curvature of the function, the absolute value of the second derivative is used to compare the nonlinearity of different regions in the dataset.

2.3. Training the proposed RBFNN

After computing the centers and the widths of the RBFNN, the network is trained by, first, using the OSD learning method to calculate the connection weights between hidden and the output layers of the network. OSD uses an optimum learning rate in each iteration of the training process [16] as follows.

Let us consider the following definitions:

$$Y_d = [y_{di}], \quad i = 1, \dots, M \quad (20)$$

where Y_d is the data sample vector and M is the number of samples $W = [w_j], \quad j = 1, \dots, N_h$ (21)

where W is the weight vector and N_h is the number of hidden neurons

$$F = [\phi_j(x_i)], \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N_h \quad (22)$$

where ϕ is the general RBF value matrix, which for the Gaussian RBFs we have

$$\phi_j(x_i) = e^{-(x_i - c_j)^2 / \sigma_{ij}^2} \quad (23)$$

In a RBF neural network, we have

$$Y = [y_i] = W\phi^T, \quad i = 1, \dots, M \quad (24)$$

where Y is the estimated output vector. It is obvious that the error vector is

$$E = Y_d - Y = Y_d - W\phi^T, \quad (25)$$

and the sum squared error, which should be minimized through the learning process, will be

$$J = \frac{1}{2} EE^T \quad (26)$$

In the conventional SD method, the new weights are computed using the gradient of J in the W space

$$OJ = \frac{\partial J}{\partial W} = \frac{\partial((1/2)EE^T)}{\partial W} = (Y_d - Y) \frac{\partial Y}{\partial W} = E \frac{\partial(W\phi^T)}{\partial W} = E\phi \quad (27)$$

$$\Delta W = OJ = E\phi \quad (28)$$

$$W_{new} = W_{old} + \lambda \Delta W, \quad (29)$$

where the coefficient λ is called learning rate (LR), and remains constant through the learning process. It is clear that although the Eq. (28) shows the optimum direction of delta weight vector, in the sense of first order estimation, but it still does not specify the optimum length of this vector; and therefore, the optimum learning rate (OLR). To achieve the OLR, the sum-squared error of the new weights should be computed employing Eqs. (25)–(27)

$$\begin{aligned} J(W + \lambda \Delta W) &= \frac{1}{2} [Y_d - (W + \lambda \Delta W)\phi^T][Y_d - (W + \lambda \Delta W)\phi^T]^T \\ &= \frac{1}{2} (E - \lambda \Delta W \phi^T)(E - \lambda \Delta W \phi^T)^T \\ &= \frac{1}{2} EE^T - \lambda E\phi \Delta W^T + \frac{1}{2} \lambda^2 \Delta W \phi^T \phi \Delta W^T \\ &= A + B\lambda + C\lambda^2, \end{aligned} \quad (30)$$

where $A = \frac{1}{2} EE^T$, $B = -E\phi \Delta W^T$ and $C = \frac{1}{2} \Delta W \phi^T \phi \Delta W^T$ are scalar constants. Thus, $J(W + \lambda \Delta W)$ is a quadratic function of λ with coefficients A , B and C . Now, considering these coefficients in detail

$$A = \frac{1}{2} EE^T = \frac{1}{2} \sum_{i=1}^M E_i^2 > 0,$$

$$B = -E\phi \Delta W^T \xrightarrow{\text{Eq. (14)}} B = -E\phi \phi^T E^T = -(E\phi)(E\phi)^T \leq 0$$

$$C = \frac{1}{2} E \phi \phi^T \phi \phi^T E^T C = (1/2) \Delta W \phi^T \phi \Delta W^T \xrightarrow{\quad} C = \frac{1}{2} (E \phi \phi^T) (E \phi \phi^T)^T \geq 0, \quad (31)$$

$J(\lambda)$ will define a quadratic function of ϕ with positive coefficients of second order term. Thus, it would have a minimum which can be found computing the derivative of $J(\lambda)$

$$\frac{\partial J}{\partial \lambda} = \frac{\partial (A + B\lambda + C\lambda^2)}{\partial \lambda} = B + 2\lambda C = 0 \xrightarrow{\text{hence}} \lambda_{\min} = -\frac{B}{2C} = -\frac{(E\phi)(E\phi)^T}{(E\phi\phi^T)(E\phi\phi^T)^T} \quad (32)$$

This LR minimizes the $J(\lambda)$, and so we can call it the OLR

$$\lambda_{\text{opt}} = \frac{(E\phi)(E\phi)^T}{(E\phi\phi^T)(E\phi\phi^T)^T} \geq 0 \quad (33)$$

Now the optimum delta weight vector (ODWV) can be determined as

$$\Delta W_{\text{opt}} = \lambda_{\text{opt}} \Delta W = \frac{(E\phi)(E\phi)^T (E\phi)}{(E\phi^T)(E\phi^T)^T} \quad (34)$$

hence

$$W_{\text{new}} = W_{\text{old}} + \frac{(E\phi)(E\phi)^T (E\phi)}{(E\phi\phi^T)(E\phi\phi^T)^T} \quad (35)$$

which the initial value for W is chosen randomly.

Table 3

Description and values of the parameters used in the improved PSO–OSD.

Parameters	Description	Considered value
k_1	Coefficient of the speed of evolution	0.2
k_2	Coefficient of the average fitness variance	0.4
ω_0	inertia factor	1
C1	Local search coefficient	1.5
C2	Social search coefficient	1.5
Population size	Number of particles	15
k_{max}	Number of iterations	400
Number of runs	–	50

Table 4

Classification error and precision (%) for the improved PSO–OSD and the PSO–OSD when using improved PSO algorithm only.

Data set	Improved PSO–OSD						PSO–OSD					
	Training set		Testing set		Precision (%)	Time (s)	Training set		Testing set		Precision (%)	Time (s)
	Ave.	S.D.	Ave.	S.D			Ave.	S.D.	Ave.	S.D.		
Iris	1.67	1.18	4.34	0.98	96.8	7.77	5.33	2.99	6.40	4.75	93.5	20.78
Wine	15.90	2.33	32.90	1.67	79.9	10.83	14.37	3.14	35.89	5.48	74.8	24.23
Abalone	13.33	1.10	23.17	1.47	67.64	90.70	47.14	1.33	48.32	1.78	63.21	205.35
Ionosphere	1.67	2.10	5.51	2.39	95.6	18.15	11.62	2.25	18.75	3.05	90.65	27.69
Glass	16.17	3.02	36.17	4.34	82.24	11.86	32.34	3	39.81	7.67	79.24	18.36

Table 5

Classification error and precision (%) for the improved PSO–OSD and the PSO–OSD when using the proposed width adjustment only.

Data set	Improved PSO–OSD						PSO–OSD					
	Training set		Testing set		Precision (%)	Time (s)	Training set		Testing set		Precision (%)	Time (s)
	Ave.	S.D.	Ave.	S.D.			Ave.	S.D.	Ave.	S.D.		
Iris	2.22	1.86	4.22	1.63	95.6	17.34	5.33	2.99	6.40	4.75	93.5	20.78
Wine	23.82	4.01	31.01	2.38	77.3	22.69	14.37	3.14	35.89	5.48	74.8	24.23
Abalone	21.90	1.27	41.87	1.59	66.7	190.01	47.14	1.33	48.32	1.78	63.21	205.35
Ionosphere	2.34	2.25	8.23	1.61	93.4	23.94	11.62	2.25	18.75	3.05	90.65	27.69
Glass	18.61	3.71	38.34	5.37	81.32	15.71	32.34	3	39.81	7.67	79.24	18.36

3. Proposed model for object image retrieval

In this section we describe different steps of our proposed method for object retrieval. In some previous works, image retrieval has been performed using image classification [29–33]. These methods extract features from the images in different categories which are then learnt using a classifier. In a next step, user enters the query image and the trained classifier predicts the class of the query image. The most similar images are then retrieved from the predicted category. In our proposed method, we detect the main object of the image before the feature extraction step. Thus, our object retrieval method composed of three main steps: (1) Main Object Detection, (2) Image Feature Extraction, and (3) Object Retrieval which is shown in Fig. 1.

3.1. Main object detection

The existing images on the web mostly contain complex backgrounds. Therefore, we propose an image segmentation as a preprocessing step in order to remove the background from an image which leads to an enhancement in the classification performance.

3.1.1. Image segmentation

Various image segmentation methods have been introduced in the literature. In our work, we use the method proposed by Achanta et al. in [34] because it is fast and efficient in terms of segmentation accuracy and memory usage.

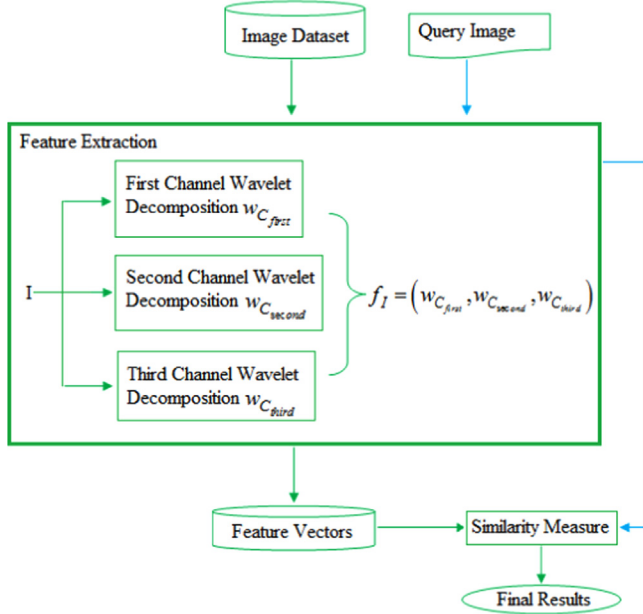
The proposed method, simple linear iterative clustering (SLIC), is based on a new approach for constructing superpixels.

This new approach generates superpixels by clustering pixels based on their color similarity and proximity in the image plane. This is done in a 5-dimensional $[l, a, b, x, y]$ space, where $[l, a, b]$ is the pixel color vector in CIELAB color space, which is widely considered as perceptually uniform for small color distances, and xy are the coordinates of the pixels. It has a different distance measurement which enables compactness and regularity in the superpixel shapes, and can be used on both grayscale and color

Table 6

Classification error and precision (%) for the Improved PSO–OSD and the PSO–OSD.

Data set	Improved PSO–OSD						PSO–OSD					
	Training set		Testing set		Precision (%)	Time (s)	Training set		Testing set		Precision (%)	Time (s)
	Ave.	S.D.	Ave.	S.D			Ave.	S.D.	Ave.	S.D.		
Iris	1.20	0.59	2.97	0.66	98.9	6.19	5.33	2.99	6.40	4.75	93.5	20.78
Wine	21.12	2.83	28.67	1.37	83.3	8.57	14.37	3.14	35.89	5.48	74.8	24.23
Abalone	25.43	1.34	28.19	1.69	72.4	79.34	47.14	1.33	48.32	1.78	63.21	205.35
Ionosphere	2.83	1.39	4.52	0.86	98.6	13.39	11.62	2.25	18.75	3.05	90.65	27.69
Glass	10.39	1.36	28.46	1.12	88.4	9.93	32.34	3	39.81	7.67	79.24	18.36

**Fig. 4.** The proposed algorithm for image retrieval.

images. SLIC takes a desired number of approximately equally-sized superpixels K as an input, and as a result each superpixel will have approximately N/K pixels. Hence, for equally sized superpixels, there would be a superpixel center at every grid interval $S = \sqrt{N/K}$. K superpixel cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ with $k = [1, K]$ at regular grid intervals S are chosen. Since the spatial extent of any cluster is approximately S^2 , it can be assumed that pixels associated with this cluster lie within 25×25 area around the superpixel center in the xy plane.

Euclidean distances in CIELAB (*Lab*) color space are meaningful for small distances. If spatial pixel distances exceed this perceptual color distance limit, they begin to outweigh the pixel color similarities. Therefore, instead of using a simple Euclidean norm in the 5D space, a distance measure D_s is used and defined as follows [34]:

$$\begin{aligned}
 d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\
 d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\
 D_s &= d_{lab} + \frac{m}{S} d_{xy}
 \end{aligned} \quad (36)$$

where D_s is the sum of the lab distance and the xy plane distance normalized by the grid interval S . A variable m is introduced in D_s allowing us to control the compactness of superpixel. The greater the value of m , the more spatial proximity is emphasized and the more compact the cluster. This value can be in the range [1, 20]. We choose $m=10$ according to the authors in the reference paper. The

Table 7

The Average precision of the methods.

Semantic name	Average precision (%)				
	RGB-Wavelet	HSV-Wavelet	YUV-Wavelet	YCbCr-Wavelet	Lab-Wavelet
African people	51.34	56.68	38.29	53.26	58.73
Beach	42.52	47.67	35.27	44.59	48.94
Building	46.41	49.57	38.21	41.36	53.74
Buses	86.54	92.68	80.69	88.45	95.81
Dinosaurs	92.68	97.21	89.84	94.54	98.36
Elephants	61.24	61.75	46.27	65.19	64.17
Flowers	72.34	81.37	69.44	77.15	85.64
Horses	67.68	77.92	66.37	69.14	80.31
Mountains and glaciers	37.27	49.17	27.26	41.58	54.27
Food	59.32	61.73	51.27	62.14	63.14
Average accuracy	61.787	67.575	54.321	63.74	70.311

Table 8

The average precision and recall of the proposed model compared with other methods.

Semantic name	Average precision (%)			
	Proposed model	Model proposed in [37]	Model proposed in [38]	Model proposed in [39]
African people	58.73	42.25	42.40	68.30
Beach	48.94	39.75	44.55	54.00
Building	53.74	37.35	41.05	56.15
Buses	95.81	74.10	85.15	88.80
Dinosaurs	98.36	91.45	58.65	99.25
Elephants	64.17	30.40	42.55	65.80
Flowers	85.64	85.15	89.75	89.10
Horses	80.31	53.80	58.90	80.25
Mountains and glaciers	54.27	29.25	26.80	52.15
Food	63.14	36.95	42.65	73.25
Average	70.311	52.64	53.24	72.70

SLIC segmentation begins by sampling K regularly spaced cluster centers and moving them to seed locations corresponding to the lowest gradient position in a 3×3 neighborhood. This is done to avoid placing them at an edge which in turn reduces the chance of choosing noisy pixels. In this method, the image gradients are computed as

$$G(x, y) = \|I(x+1, y) - I(x-1, y)\|^2 + \|I(x, y+1) - I(x, y-1)\|^2 \quad (37)$$

where $I(x, y)$ is the *Lab* vector corresponding to the pixel at position (x, y) , and $\|\cdot\|$ is the L_2 norm. This takes into account both color and

intensity information. Each pixel in the image is then associated with a nearest cluster center whose search area overlaps the pixel. After associating all the pixels, new centers are computed as the average *labxy* vectors of the assigned pixels to the clusters.

At the end of this process, a few stray labels may remain near to a large segment which have been assigned to a same cluster but not connected to the segment. It enforces connectivity in the last step of the algorithm by assigning the disjoint segments to the largest neighboring cluster. The following algorithm shows the steps of SLIC segmentation [34].

object region from the center of the image. For this reason, we consider a half window of the image as the region of interest (the blue window in the segmented image of Fig. 2). The largest region within this window is most likely to be a part of the main object; therefore, it is selected as the candidate region of the main object (the region is highlighted in blue in Fig. 2).

3.1.3. Object region and background region labeling

In order to discriminate the background and the main object regions, color and texture features are extracted from every region.

Algorithm 3. The algorithm of the proposed width adjustment.

- 1: Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ sampling pixels at regular grid steps S .
- 2: Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position.
- 3: **for** each cluster center C_k **do**
- 4: Assign the best matching pixels from a $2S \times 2S$ square neighborhood around the cluster center according to the distance measure (Eq. (36)).
- 5: **end for**
- 6: Compute new cluster centers and residual error E (L_1 distance between previous centers and recomputed centers)
- 7: Enforce connectivity.

3.1.2. Main region detection

After image segmentation in order to capture the most relevant features from the image, the main object of the image should be separated from the background of the image. Since the main object usually appears near the center of the image, we extract the main

Then we compare feature vectors of the regions to the feature vector of the main object region which allows us to label the regions as background or main object. We use the first two statistical color moments as color feature, and Tamura features, namely coarseness, as texture features to detect similar regions to the main object region. The first two color moments are computed



Fig. 5. The retrieval results of the horse image using wavelet features in *Lab* color space.

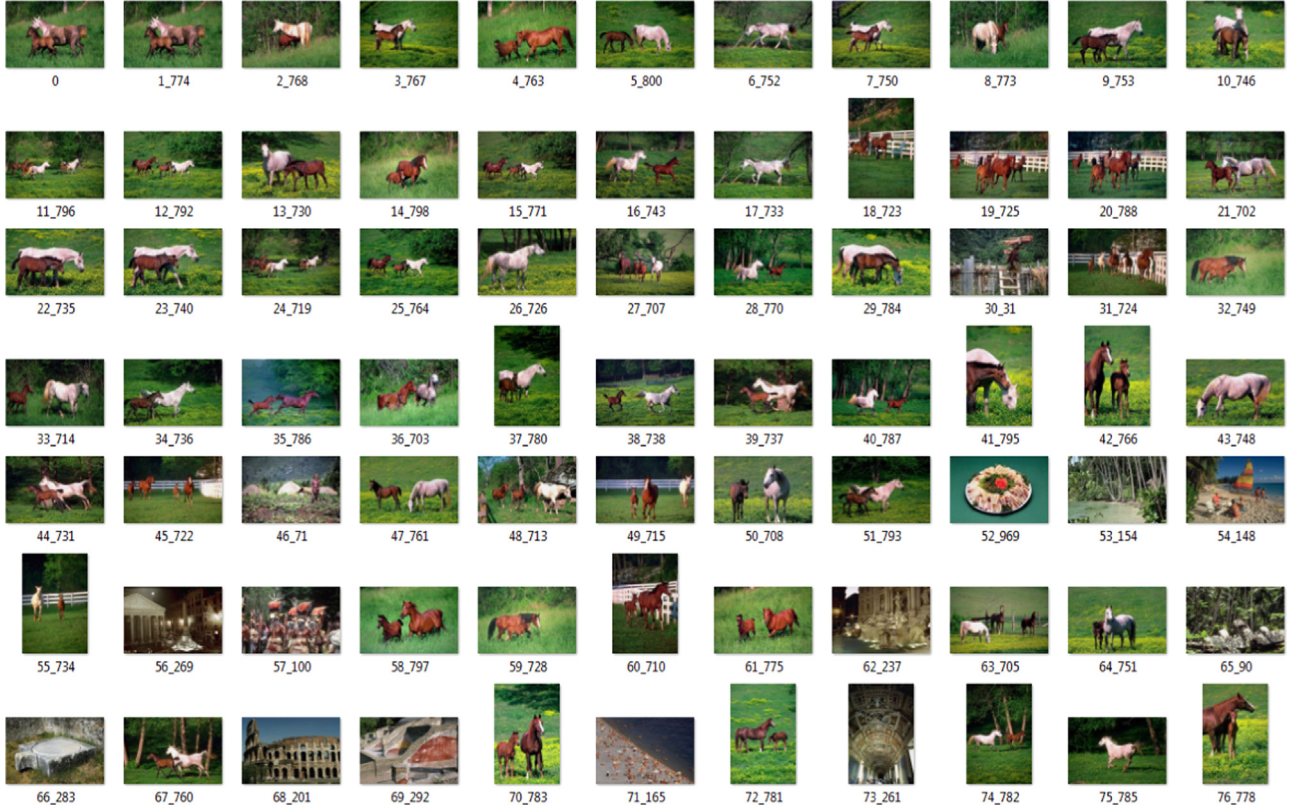


Fig. 6. The retrieval results of the horse image using wavelet features in HSV color space.

as follows:

$$\mu_i = \frac{1}{N} \sum_{j=1}^N f_{ij}, \quad \sigma_i = \left(\frac{1}{N} \sum_{j=1}^N (f_{ij} - \mu_i)^2 \right)^{1/2}, \quad i = 1, 2, 3. \quad (38)$$

where f_{ij} is the value of the i th color component of the image pixel “ j ”, and N is the total number of pixels in the image.

Coarseness is a measure of the granularity of a texture which relates to distances of notable spatial variations of grey levels, that is, implicitly, to the size of the primitive elements (texels) forming the texture. To compute the coarseness, moving averages $A_k(x, y)$ are computed first using $2^k \times 2^k$ ($k = 0, 1, \dots, 5$) size windows at each pixel (x, y) :

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} g(i, j) / 2^{2k} \quad (39)$$

where $g(i, j)$ is the pixel intensity at (i, j) . Then the differences between pairs of non-overlapping moving averages in the horizontal and vertical directions for each pixel are computed

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|$$

$$E_{k,v}(x, y) = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})| \quad (40)$$

After that, the value of k that maximizes E in either direction is used to set the best size for each pixel:

$$S_{best}(x, y) = 2^k \quad (41)$$

the coarseness is then computed by averaging S_{best} over the entire image

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n S_{best}(i, j) \quad (42)$$

then by using Euclidean distance measure the similarity between color and texture features of the main object region (FV_{MO}) and other regions (FV_{Other}) is calculated as the following:

$$Sim(FV_{MO}, FV_{Other}) = \sqrt{\sum_{i=1}^9 (FV_{MO_i} - FV_{Other_i})^2}$$

$$FV_{MO} = (R_{\mu}^{MO}, R_{\sigma}^{MO}, R_{crs}^{MO}, G_{\mu}^{MO}, G_{\sigma}^{MO}, G_{crs}^{MO}, B_{\mu}^{MO}, B_{\sigma}^{MO}, B_{crs}^{MO})$$

$$FV_{Other} = (R_{\mu}^O, R_{\sigma}^O, R_{crs}^O, G_{\mu}^O, G_{\sigma}^O, G_{crs}^O, B_{\mu}^O, B_{\sigma}^O, B_{crs}^O) \quad (43)$$

After extracting feature vectors from all of the regions, the regions that are not similar to the main object regions are regarded as background regions. In Fig. 2 these regions are depicted using red squares. In addition, all corner regions, depicted by yellow squares, are considered as background.

3.2. Image feature extraction

Discrete Wavelet Transform [35] is currently used in a wide variety of signal processing applications, such as image, audio and video compression, removal of noise in audio, and the simulation of wireless antenna distribution. Discrete Wavelet decomposition of image produces the multi-resolution representation of image. A multi-resolution representation provides a simple hierarchical framework for interpreting the image information. At different resolutions, the details of an image generally characterize different physical structures of the image. At a coarse resolution, these details correspond to the larger structures which provide the

Table 9

Parameter selection of the improved PSO–OSD.

Data set	No. of hidden neurons	No. of epochs
Caltech 101	675	4,500,000
Description and values of the parameters used in the improved PSO–OSD		
Parameters	Description	Considered value
k_1	Coefficient of the speed of evolution	0.2
k_2	Coefficient of the average fitness variance	0.4
ω_0	inertia factor	0.6
C1	Local search coefficient	1.5
C2	Social search coefficient	1.5
Population size	Number of particles	70
k_{max}	Number of iterations	600
Number of runs	–	30

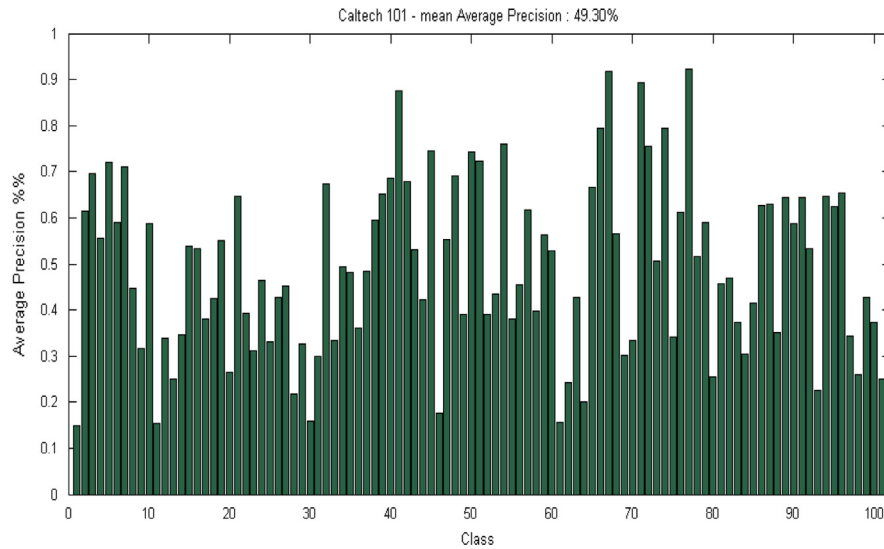
**Fig. 7.** Classification accuracy of the improved PSO–OSD on entire images.

image context. The following section briefly reviews the Two Dimensional Wavelet Transformation. The original image I is thus represented by a set of sub images at several scales; $\{\{Ld, Dnl\} | l, \dots, n, \dots, d\}$, which is multi-scale representation with depth d of the image I . The image is represented by two dimensional signal functions; wavelet transform decomposes the image into four frequency bands, namely the **LL1**, **HL1**, **LH1** and **HH1** bands. **H** and **L** denote the high pass and low pass filters, respectively. The approximated image **LL** is obtained by low pass filtering in both row and column directions. The detailed images **LH**, **HL** and **HH** contain the high frequency components. To obtain the next coarse level of wavelet coefficients, the subband **LL1** alone is further decomposed and critically sampled. Similarly **LL2** will be used to obtain further decomposition. Decomposing the approximated image at each level into four sub-images forms the pyramidal image tree. This results in two-level wavelet decomposition of image as shown in Fig. 3.

In all of the experiments in this paper we used the HH1 domain that contains more information than others.

3.3. Object image retrieval

After main object detection, we extract object features and construct feature vectors using wavelet transform in CIELAB (*Lab*) color space. To do so, we apply wavelet transform on each color channel namely L , a and b . Therefore, for the main object image I in

the dataset, its feature vector f_I is defined as follows:

$$f_I = (f_L, f_a, f_b) \quad (44)$$

where f_L is the feature vector extracted from the main object image in channel L , f_a is the feature vector extracted from the main object image in channel a and f_b is feature vector extracted from the main object image in channel b . Each of the feature vectors is of dimension 40, so the feature vector f_I is in 120-dimensional space. After this step, extracted features are fed into the improved PSO–OSD radial basis function neural network to train the network. Once training was done, the trained network is able to predict the class of the query image. By determining the class of the query image, the most similar images are shown to the user (Fig. 1).

4. Experimental results

In this section, we first give the results of improving the Three-Phased PSO–OSD neural networks (PSO–OSD for short) by studying the impact of improving the PSO algorithm and the impact of the proposed method for the width adjustment of the RBFNN, separately.

Then we experimentally show that applying wavelet transform on images in *Lab* color space leads higher performance compared to other color spaces like *RGB*, *HSV*, *YUV* and *YCbCr*. Finally, we

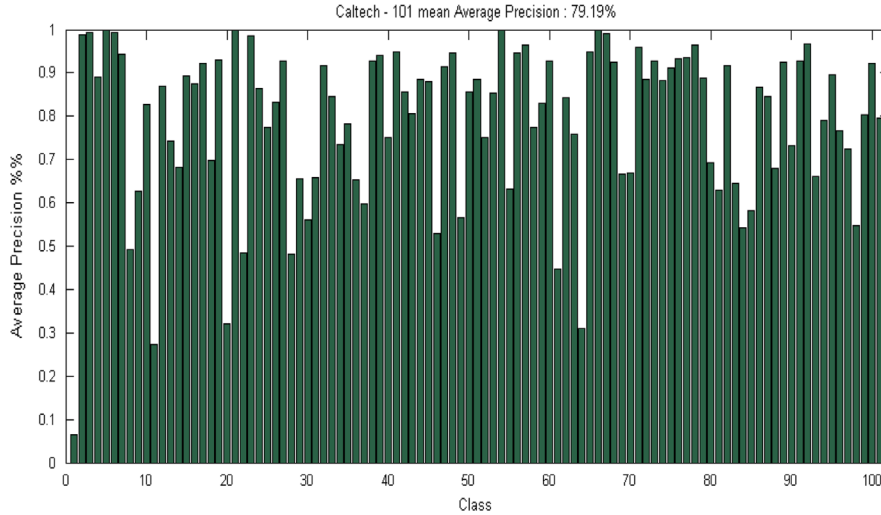


Fig. 8. Classification accuracy of the improved PSO-OSD on object images.

Table 10

Performance evaluation on the Caltech 101 dataset.

Approach	Classification accuracy (%)
Method proposed in [41]	76.4
Method proposed in [42]	73.44
Method proposed in [43]	73.20
Method proposed in [44]	77.8
Method proposed in [45]	75.7
Method proposed in [46]	64.6
Our proposed method	79.19

report the results of object image retrieval using the proposed method.

4.1. Experimental results of improving the PSO-OSD RBFNN

We test the improved PSO-OSD on five benchmark problems from Proben1 (Prechelt1994) dataset in the UCI machine learning repository to have a fair comparison with the PSO-OSD method. Table 1 gives information about the dataset. Both PSO based methods are used in the OSD algorithm for estimating the connection weights. For classification problems, the output of the network with the highest response is taken and the corresponding class is considered as the winning class for the input vector. All experiments have been run 50 times and in all of them, 50% of the dataset is used as the train set and the rest is considered as test set to validate the functionality of trained network.

As stated before, the structure of the RBFNN has three layers. The number of neurons in each layer affects the network performance. It should be noted that the number of neurons in the hidden layer and optimum values of different PSO parameters have been tuned through conducting various experiments. The values and the short description of the parameters used in the algorithms are shown in Tables 2 and 3.

For comparison, statistical results such as classification error and precision are reported in Tables 4–6. In Table 4, we consider the effect of improving PSO algorithm only. In addition, we report the run time of the improved PSO-OSD RBFNN and the PSO-OSD RBFNN. The run time is the total time needed for 50 runs of the algorithm (training and testing the RBFNN).

Table 4 states that by adjusting centers using the proposed PSO algorithm, the performance of the RBFNN improves while being faster, as well.

Table 5 shows the effect of adjusting the widths of the PSO-OSD using our proposed method for width determination only. It can be seen from this table that the proposed method achieves higher performance compared to the PSO-OSD although the training time of two RBFNNs is almost the same.

From Tables 4 and 5, first of all, it can be concluded that both the proposed methods works well and improves the performance of the PSO-OSD RBFNN. Next, the proposed method for improving the PSO algorithm is more effective in comparison with the proposed strategy for width adjustment from both viewpoints namely increasing the classification performance of the PSO-OSD RBFNN and decreasing the training time of the PSO-OSD RBFNN.

Table 6 reports the results of combining the two improvements together. From this table we see that combining two improvements also increases the performance and decreases the training time of the proposed RBFNN.

As shown in this table, improved PSO-OSD achieves better results than PSO-OSD. Using the improved PSO-OSD algorithm on test set of Iris dataset, the precision is 98.9% while the PSO-OSD results in the precision of 93.5%. Further, on other datasets the improved PSO-OSD RBFNN increases the performance of the PSO-OSD by 8.5%, 9.19%, 7.95% and 9.16%. Therefore, on average the proposed method enhances the precision of the method in [5] by 8.04%. Based on the statistics, standard deviations of the improved PSO-OSD in Tables 4–6, reached to smaller values, which mean that it has better repeatability for the retrain process than PSO-OSD method. For all datasets, improved PSO-OSD decreased the standard deviation of classification error for the test sets (these values have been shown in bold).

4.2. Experimental results on different color spaces

In this section, we study that applying wavelet transform in *Lab* color space leads the most promising results in image retrieval. To this end, we use Corel dataset [36]. This dataset consists of 1000 images in 10 categories (1: African people and village, 2: Beach, 3: Building, 4: Buses, 5: Dinosaurs, 6: Elephants, 7: Flowers, 8: Horses, 9: Mountains and glaciers, 10: Food) and 100 images in each category.

To build an image retrieval model, we apply the proposed algorithm in Fig. 4. As it can be seen from this figure, first of all we convert all the images from *RGB* color space to different color spaces, namely *HSV*, *YUV*, *YCbCr* and *Lab*. Then the images in each color space are decomposed to their channels. For instance in *YUV* color space the images are decomposed to channels *Y*, *U* and *V*.

Next, wavelet transform is applied on each color channel and features are extracted from those channels and stored. Once the query image is entered its feature vector is extracted and using similarity measure unit, the similarity of the query image and images in the dataset is determined using L_2 distance measure and the most similar images are returned as the query results. In order to compare the effect of the color spaces on the performance of the proposed CBIR model, statistical measures such as Precision and Recall are reported in Table 7. These measures are computed as follows:

$$\text{precision} = \frac{tp}{tp + fp}$$

$$\text{recall} = \frac{tp}{tp + fn} \quad (45)$$

Considering this table, the most promising results are achieved using *Lab* color space. After *Lab* color space *HSV* color space achieves the best results.

This table shows that wavelet features extracted from the images in *Lab* color space achieve better performance compared to the other color spaces. That is our reason for applying wavelet transform in *Lab* color space on main object images. Although the proposed method is simple, it achieves promising results compared to other proposed methods in [37–39] (Table 8).

Fig. 5 shows the retrieval results of a horse image as the query image using the proposed model in *Lab* color space while Fig. 6 shows the retrieval results of that image using the proposed model in *HSV* color space. It should be noted that in both figures the results of the first 75 similar images are presented.

4.3. Experimental results on Caltech 101

In this section, we report results of the implementation of the proposed method on Caltech 101 dataset [40]. The Caltech 101 dataset is composed of 9144 images split into 101 object categories, such as tools, artifacts, and animals, as well as one background category with significant variances in shape. The number of images in each class varies from 31 to 800. In the experiments, the images are resized to no larger than 300×300 pixels with a preserved aspect ratio for computational efficiency. In our experiments in order to demonstrate the effectiveness of the main object extraction, we first compare the performance of the image retrieval by using features extracted from the entire images and features extracted from the main object images.

In these experiments the improved PSO–OSD RBFNN is used to determine the class of the query image. The parameters for training the RBFNN are reported in Table 9.

Fig. 7 shows the mean average precision of each category while features extracted from the entire images and Fig. 8 shows the mean average precision of each category while features extracted from the main objects.

Considering this results, we observe that when features extracted from the main object, the performance increases from 49.30% to 79.19%. For all the categories except category number 1, which is Google Background, precision has been notably improved.

For fair comparisons, as suggested by the original dataset [40] and also by many others [41–47], the whole dataset is partitioned into 5, 10, ..., 30 training images per class and no more than 50 testing images per class, and the performance is measured using average accuracy over 102 classes (i.e. 101 classes and a “background” class). Here we use 30 training images per class and no more than 50 testing images per class so that we can compare our results with the results reported in the literature.

Table 10 shows experimental comparison between our proposed method and other methods. From this table, we can see that

the proposed approach outperforms several existing approaches [41–46].

In our evaluation, totally 4 classes achieve 100% classification accuracy, which are Motorbikes, Car side, Inside skate and Metro-nome. Also, 32 classes achieve classification accuracies higher than 90%. In addition, there are 6 classes that their classification accuracies are less than 0.35%.

The final classification result which is 79.19% states that extracting image features from the main object image considerably increases the performance while the improved PSO–OSD RBFNN is employed as our classifier.

5. Conclusion

This paper proposed two significant improvements for the Three-Phased PSO–OSD RBFNN. The first improvement introduced a new version of PSO algorithm for determining the centers of the RBFNN units which increased the both global and local search abilities of the PSO as well as its convergence speed. Then a new method for determining the widths of the RBFNN was proposed in which spatial information of the data and nonlinearity of the function to be approximated were taken into account. The experimental results showed that applying each of the improvements separately caused increasing the performance of the classifier. Moreover, using both these improvements we showed that the performance of the RBFNN was further increased. To show the ability of the proposed PSO–OSD RBFNN, we test it on five benchmark datasets and object image retrieval problem. For the object image retrieval problem, we introduced a new idea for main object detection using SLIC segmentation. In addition, we experimentally demonstrated that wavelet features extracted from *Lab* color space were more efficient than wavelet features extracted from other color spaces.

Acknowledgments

The authors are grateful to the anonymous reviewers for the insightful comments and constructive suggestions. Part of this research was funded by Iranian Research Institute for Information Science and Technology (IranDoc) (No. TMU92-03-44).

References

- [1] D.S. Broomhead, D. Lowe, Multi-variable functional interpolation and adaptive networks, *Complex Syst.* 2 (1988) 321–355.
- [2] M. Powell, Radial basis functions for multivariable interpolation: a review, in: J. C. Mason, M.G. Cox (Eds.), *Algorithms for Approximation*, Clarendon Press, New York, NY, USA, 1987, pp. 143–167.
- [3] N. Dyn, C. Micchelli, Interpolation by sums of radial functions, *Math* 58 (1990) 1–9.
- [4] J. Moody, C. Darken, Fast learning in networks of locally-tuned processing units, *Neural Comput.* 1 (2) (1989) 281–294. <http://dx.doi.org/10.1162/neco.1989.1.2.281>.
- [5] V. Fathi, G.A. Montazer, Three-phase strategy for the osd learning method in rbf neural networks, *Neurocomputing* 111 (2013) 169–176. <http://dx.doi.org/10.1016/j.neucom.2012.12.024>.
- [6] I. Lin, C. Liou, Least-mean-square training of cluster-weighted modeling, in: *Artificial Neural Networks—ICANN 2007, Lecture Notes in Computer Science*, vol. 4669, 2007, pp. 301–310, http://dx.doi.org/10.1007/978-3-540-74695-9_31.
- [7] R. Neruda, P. Kudova, Learning methods for radial basis function networks, *Neural Comput.* 21 (7) (2005) 1131–1142. <http://dx.doi.org/10.1016/j.future.2004.03.013>.
- [8] T. Poggio, F. Girosi, Networks for approximation and learning, *Proc. IEEE* 78 (9) (1990) 1481–1497, <http://dx.doi.org/10.1109/5.58326>.
- [9] C. Bishop, Improving the generalization properties of radial basis function neural networks, *Neural Comput.* 3 (4) (1991) 579–588.
- [10] K.K. Tan, K.-Z. Tang, Adaptive online correction and interpolation of quadrature encoder signals using radial basis functions, *IEEE Trans. Control Syst. Technol.* 13 (3) (2005) 370–377. <http://dx.doi.org/10.1109/TCST.2004.841648>.

- [11] C.F.N.C.S. Chen, S.A. Billings, P.M. Grant, Practical identification of narmax models using radial basis functions, *Int. J. Control* 52 (6) (1990) 1327–1350. <http://dx.doi.org/10.1080/00207179008953599>.
- [12] T. Mu, A. Nandi, Detection of breast cancer using v-svm and rbf networks with self organized selection of centers, in: Third IEEE International Seminar on Medical Applications of Signal Processing, 2005, pp. 47–52.
- [13] A.M.M. Orr, J. Hallam, T. Leonard, Assessing rbf networks using delve, *Int. J. Neural Syst.* 10 (5) (2000) 397–415.
- [14] X. Chen, Least-mean-square training of cluster-weighted modeling, in: Advances in Neural Networks—Lecture Notes in Computer Science, vol. 4493, 2007, pp. 41–48. http://dx.doi.org/10.1007/978-3-540-72395-0_6.
- [15] R. Cancelliere, M. Gai, A comparative analysis of neural network performances in astronomical imaging, *Appl. Numer. Math.* 45 (1) (2003) 87–98.
- [16] R.S.G.A. Montazer, H. Khatir, Improvement of learning algorithms for rbf neural networks in a helicopter sound identification system, *Neurocomputing* 71 (1–3) (2007) 167–173.
- [17] R.S.G.A. Montazer, F. Ghorbani, Three-phase strategy for the osd learning method in rbf neural networks, *Neurocomputing* 72 (7–9) (2009) 1797–1802.
- [18] V. Balamurugan, A.Kumar, An integrated color and texture feature based framework for content based image retrieval using 2d wavelet transform, in: International Conference on Computing, Communication and Networking, 2008, pp. 1–16. <http://dx.doi.org/10.1109/ICCCNET.2008.4787734>.
- [19] G.C.B.C.G. Quéllec, M. Lamard, C. Roux, Fast wavelet-based image characterization for highly adaptive image retrieval, *IEEE Trans. Image Process.* 21 (4) (2012) 1613–1623. <http://dx.doi.org/10.1109/TIP.2011.2180915>.
- [20] A.K.S. Agarwal, P. Singh, Content-based image retrieval using discrete wavelet transform and edge histogram descriptor, in: International Conference on Information Systems and Computer Networks (ISCON), 2013, pp. 19–23. <http://dx.doi.org/10.1109/ICISCON.2013.6524166>.
- [21] Y. Wang, W. Zhang, Coherence vector based on wavelet coefficients for image retrieval, in: IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 2, 2012, pp. 765–768. <http://dx.doi.org/10.1109/CSAE.2012.6272878>.
- [22] M. Yildizer, A.M. Balci, R. Alhaji, Efficient content-based image retrieval using multiple support vector machines ensemble, *Expert Syst. Appl.* 39 (3) (2012) 2385–2396. <http://dx.doi.org/10.1016/j.eswa.2011.08.086>.
- [23] R.S.Gh.A. Montazer, F. Ghorbani, Improvement of learning rate for rbf neural networks in a helicopter sound identification system introducing two-phase osd learning method, in: Proceeding of the Fifth International Symposium on Mechatronics and its Applications, 2008, pp. 1–5. <http://dx.doi.org/10.1109/ISMA.2008.4648802>.
- [24] H.K.Gh.A. Montazer, V. Fathi, Improvement of rbf neural networks using fuzzy-osd algorithm in an online radar pulse classification system, *Appl. Soft Comput.* 13(9) (2013) 3831–3838. <http://dx.doi.org/10.1016/j.asoc.2013.04.021>.
- [25] A. Sanchez, V. David, Second derivative dependent placement of rbf centers, *Neurocomputing* 7 (3) (1995) 311–317. [http://dx.doi.org/10.1016/0925-2312\(94\)00082-4](http://dx.doi.org/10.1016/0925-2312(94)00082-4).
- [26] Y.W.S. Chen, B.L. Luk, Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks, *IEEE Transactions on Neural Networks* 10(5) (1999) 1239–1243. <http://dx.doi.org/10.1109/72.788663>.
- [27] Y.W.M.X.B. Shi, L. Yu-xia, Y. Xin-hua, A modified particle swarm optimization and radial basis function neural network hybrid algorithm model and its application, in: IEEE Computer Society Global Congress on Intelligent Systems, vol. 1, 2009, pp. 134–138. <http://dx.doi.org/10.1109/GCIS.2009.233>.
- [28] N. Benoudjit, M. Verlysen, On the kernel widths in radial-basis function networks, *Neural Process. Lett.* 18 (2) (2003) 139–154. <http://dx.doi.org/10.1023/A:1026289910256>.
- [29] W.-T. Wong, S.-H. Hsu, Application of svm and ann for image retrieval, *Eur. J. Oper. Res.* 173 (3) (2005) 938–950. <http://dx.doi.org/10.1016/j.ejor.2005.08.002>.
- [30] J.V.-C.J. Félix Serrano-Talamantes, C. Avilés-Cruz, J.H. Sossa-Azuela, Self organizing natural scene image retrieval, *Expert Syst. Appl.* 40 (7) (2013) 2398–2409.
- [31] K.-K.S. S-Sung Park, D.-S. Jang, Expert system based on artificial neural networks for content-based image retrieval, *Expert Syst. Appl.* 29 (3) (2005) 589–597. <http://dx.doi.org/10.1016/j.eswa.2005.04.027>.
- [32] B.M.S. Sadek, A. Al-Hamadi, U. Sayed, Cubic-splines neural network-based system for image retrieval, in: 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 273–276. <http://dx.doi.org/10.1109/ICIP.2009.5413561>.
- [33] J.K.D.S. Mukhopadhyay, R.D. Gupta, Content based texture image retrieval using fuzzy class membership, *Pattern Recognit. Lett.* 34 (6) (2013) 646–654. <http://dx.doi.org/10.1016/j.patrec.2013.01.001>.
- [34] A.P.F.R. Achanta, A. Shaji, K. Smith, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (11) (2012) 2274–2282. <http://dx.doi.org/10.1109/TPAMI.2012.120>.
- [35] L.N.J.M.J.A. Zegarrra, R. da Silva Torres, Wavelet-based fingerprint image retrieval, *J. Comput. Appl. Math.* 227 (2) (2009) 294–307. <http://dx.doi.org/10.1016/j.cam.2008.03.017>.
- [36] J.L.J. Wang, G. Wiederhold, Semantics-sensitive integrated matching for picture libraries, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (9) (2001) 947–963. <http://dx.doi.org/10.1109/34.955109>.
- [37] G.S.N. Jhanwara, S. Chaudhuri, B. Zavidovique, Content based image retrieval using motif co-occurrence matrix, *Image Vis. Comput.* 22 (14) (2004) 1211–1220. <http://dx.doi.org/10.1016/j.imavis.2004.03.026>.
- [38] P.W. Huang, S. Dai, Image retrieval by texture similarity, *Pattern Recognit.* 36 (3) (2003) 665–679. [http://dx.doi.org/10.1016/S0031-3203\(02\)00083-3](http://dx.doi.org/10.1016/S0031-3203(02)00083-3).
- [39] Y.-K.C.Ch.-H. Lin, R.-T. Chen, A smart content based image retrieval system based on color and texture features, *Image Vis. Comput.* 27 (6) (2009) 658–665. <http://dx.doi.org/10.1016/j.imavis.2008.07.004>.
- [40] R.F.L. Fei-Fei, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *Comput. Vis. Image Underst.* 106 (1) (2007) 59–70. <http://dx.doi.org/10.1016/j.cviu.2005.09.012>.
- [41] Z.L.L.Y.H. Cheng, R. Yu, X.wen. Chen, Kernelized pyramid nearest-neighbor search for object categorization, *Mach. Vis. Appl.* 25 (4) (2014) 931–941. <http://dx.doi.org/10.1007/s00138-014-0608-3>.
- [42] K.Y.F.L.-T.H.J. Wang, J. Yang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3360–3367. <http://dx.doi.org/10.1109/CVPR.2010.5540018>.
- [43] Y.G.J. Yang, K. Yu, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1794–1801. <http://dx.doi.org/10.1109/CVPR.2009.5206757>.
- [44] H.L.K. Sohn, D.Y. Jung, A.O. Hero, Efficient learning of sparse, distributed, convolutional feature representations for object recognition, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2643–2650. <http://dx.doi.org/10.1109/ICCV.2011.6126554>.
- [45] Y.L.Y.-L. Boureau, F. Bach, J. Ponce, Learning mid-level features for recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2559–2566. <http://dx.doi.org/10.1109/CVPR.2010.5539963>.
- [46] C.S.S. Lazebnik, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2169–2178. <http://dx.doi.org/10.1109/CVPR.2006.68>.
- [47] Gh.A. Montazer, D. Giveki, Content based image retrieval system using clustered scale invariant feature transforms, *Optik - International Journal for Light and Electron Optics* 126 (18) (2015) 1695–1699. <http://dx.doi.org/10.1016/j.jlileo.2015.05.002>.



Gh.A. Montazer received his B.Sc. degree in Electrical Engineering from Kh.N. Toosi University of Technology, Tehran, Iran, in 1991, his M.Sc. degree in Electrical Engineering from Tarbiat Modares University, Tehran, Iran, in 1994, and his Ph.D. degree in Electrical Engineering from the same university, in 1998. He is an Associate Professor in the Department Of Information Technology at School Of Engineering, Tarbiat Modares University (TMU). His areas of research include artificial intelligence, soft computing approaches such as artificial neural network (ANN), fuzzy set theory (FST) and rough set theory (RST), pattern recognition, e-learning, and e-government.



D. Giveki received his B.Sc. degree in Hardware Engineering from Sajad University of Mashhad, Mashhad, Iran, in 2007, his M.Sc. degree in Computer Science from University of Tehran, Tehran, Iran, in 2009. Then he had a one-year visit in Saarland University in Germany in 2010–2011. He started his Ph.D. in Iranian Research Institute for Information Science and Technology (IRAN-DOC) in 2012. His research areas include PDE-based image processing, artificial neural networks and evolutionary computation.