Neural Process Lett
DOI 10.1007/s11063-017-9614-6

Scene Classification Using Multi-Resolution WAHOLB Features and Neural Network Classifier

Gholam Ali Montazer[1,2] · **Davar Giveki**[2]

Abstract This article approaches scene classification problem by proposing an enhanced bag of features (BoF) model and a modified radial basis function neural network (RBFNN) classifier. The proposed BoF model integrates the image features extracted by histogram of oriented gradients, local binary pattern and wavelet coefficients. The extracted features are obtained in a hierarchical multi-resolution manner. The proposed approach is able to capture multi-level (the pixel-, patch-, and image-level) features. The histograms of features constructed by BoF model are then used for training a modified RBFNN classifier. As a modification, we propose using a new variant of particle swarm optimization, in which the parameters are updated adaptively, for determining the center of Gaussian functions in RBFNN. Experimental results demonstrate that our proposed approach significantly outperforms the state-of-the-art methods on scene classification of OT, FP, and LSP benchmark datasets.

Keywords Scene classification · Radial basis function neural network (RBFNN) classifier · Modified PSO–OSD (mPSO–OSD) · Bag of features (BoF) · Wavelet transform · HOG · LBP

1 Introduction

Computer vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of

✉ Gholam Ali Montazer
montazer@modares.ac.ir

Davar Giveki
Giveki@students.irandoc.ac.ir

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

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

Published online: 03 April 2017

these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka "deep learning") approaches have greatly advanced the performance of these state-of-the-art visual recognition systems [22].

For instance, in [26] authors trained a large, deep convolutional neural network to classify 1.2 million images into the 1000 different classes. In [8] authors proposed a new and simple version of deep convolutional neural networks for image classification. In [13] authors used deep learning for face alignment. Wang and Tao [55] applied deep learning for image restoration. In [50] authors investigated deep learning for pedestrian detection especially when complex occlusion was present. Simonyan and Zisserma [46] employed deep learning for large-scale image recognition. Wang and Yeung [57] introduced a deep learning tracker for visual tracking. Wang et al. [53] applied deep learning for multi-modal retrieval. Zhu et al. [66] used features extracted by deep learning for 3D shape retrieval.

Although deep learning has reached very promising results in many computer vision tasks, multi-view learning is a rapidly growing direction in machine learning with well theoretical underpinnings and great practical success [48]. Multi-view learning is concerned with the problem of machine learning from data represented by multiple distinct feature sets or "multiple views". These views may be obtained from multiple sources or different feature subsets. For example, a person can be identified by face, fingerprint, signature or iris with information obtained from multiple sources, while an image can be represented by its color or texture features, which can be seen as different feature subsets of the image [59].

Though there is significant variance in the approaches to integrating multiple views to improve learning performance, they mainly exploit either the consensus principle or the complementary principle to ensure the success of multi-view learning. Since accessing multiple views is the fundament of multi-view learning, with the exception of study on learning a model from multiple views, it is also valuable to study how to construct multiple views. Overall, by exploring the consistency and complementary properties of different views, multi-view learning is rendered more effective, more promising, and has better generalization ability than single-view learning [59].

Using multi-view learning the low-level features can be hand-crafted with great success for various computer vision tasks such as image classification [61]. Classifying various indoor and outdoor scenes is a big challenge due to the variability, the ambiguity, and the wide range of illumination and scale conditions of the available scene images.

In scene classification scenarios, developing methods to effectively represent images with their visual features and using efficient machine learning-based classification algorithms, are two important issues [3,58].

Local feature descriptors have been widely employed for bioengineering applications [14–17], image matching [47,64], retrieval [49,51,60], and classification [4,43] in the past years, due to their robustness against the changes in appearance and occlusion. Among the various existing local feature extraction methods, scale invariant feature transform (SIFT) [30,35], histogram of oriented gradients (HOG) [10], and local binary pattern (LBP) [34,36] are the most popular ones. In spite of their advantages, local feature descriptors cannot effectively represent the whole image in high-level visual tasks with large number of scene categories.

To tackle this problem, the bag of features model [19] has been introduced, which quantizes the local features extracted from an image into discrete visual words regardless of their order. The image is then represented as a visual word-occurrence histogram. However, disregarding spatial information of the local features usually deteriorates the classification accuracy.

In order to deal with this shortcoming of the BoF model, a number of methods have been proposed by previous works. Bosch et al. in [4] used probabilistic latent semantic analysis (pLSA), a generative model from the computer linguistic literature to discover objects in scene images in an unsupervised manner. The authors then used this object distribution to perform scene classification. Lazebnik et al. in [27] introduced a method which partitions an image into increasingly fine sub-regions, the so-called "spatial pyramid", and computes histograms of local features for each sub-region. The authors showed that using spatial pyramid considerably improved performance of scene classification. Oliva and Torralba [37] proposed a set of perceptual dimensions (naturalness, openness, roughness, expansion, and ruggedness) that represent the dominant spatial structure of a scene. These dimensions may be reliably estimated using spectral and coarsely localized information. Each scene image can be represented by a holistic description termed spatial envelope. This model achieved high accuracy in recognizing natural scenes. However, when categories of indoor scenes are included, its performance drops dramatically. Bosch et al. in [5] introduced pyramid of histograms of orientation gradients (PHOG) inspired by spatial pyramid matching and histograms of oriented gradient [10]. PHOG captures the global spatial information of images and represents them by histograms of edge orientations at multiple spatial pyramid levels.

In [60], authors studied the combinations of SIFT with LBP and HOG with LBP to achieve descriptive image representation based on the BoF model. The authors qualitatively and quantitatively showed that the combination of SIFT and LBP yields promising results especially on images with noisy background and ambiguous objects. Tian et al. [49] introduced a new feature descriptor, the so-called edge orientation difference histogram (EODH), which is rotation and scale invariant. The authors fused the EODH feature vectors to Color-SIFT feature descriptors before using in scene classification. Walia and Pal [51] improved the color difference histogram (CDH) and angular radial transform (ART) methods to represent images by their color, texture, and shape features. Zheng et al. [64] combined SIFT and rotation invariant LBP for an image matching problem, where the LBP is used to describe the local region centered at the keypoints determined by the SIFT detector.

Inspired by the previous efforts in enhancing BoF model, in this article, we propose a new image representation method, so-called WAHOLB, which improves the BoF model by introducing a combination of local feature descriptors (e.g. HOG and LBP) and preserving their spatial information. To this end, our proposed method uses wavelet transform to provide a hierarchical multi-resolution representation of images. Thus, it can extract multi-level (pixel-level, patch-level and, image-level) features.

As the second contribution of this article, we modify the radial basis function neural network (RBFNN) classifier proposed by Montazer and Fathi [18]. The authors in [18] proposed a new learning method for RBFNN, which employs a new center determination algorithm based on particle swarm optimization (PSO) technique. Although the proposed RBFNN shows significant improvement in pattern classification, it cannot escape from the local optima. This reduces the model's local and global search ability and decreases the convergence speed of the PSO algorithm for large datasets. To overcome these shortcomings, in this article, we propose a new variant of the PSO technique in which the PSO's parameters are updated adaptively. This PSO is then used to determine the centers of the Gaussian functions in the RBFNN's hidden layer.

Experimental results demonstrate the proposed classifier enhances the performance of scene classification task. The rest of the paper is organized as follows: The proposed BoF model is explained in Sect. 2. In Sect. 3, the proposed classifier is discussed. Section 4 demonstrates experimental results and conclusion is drawn in Sect. 5.

2 The Proposed BoF Model

This section discusses different steps of building bag of feature model using the proposed WAHOLB features. Therefore, at first, extracting WAHOLB features are elaborated.

To extract image features, images are divided into small overlapped patches with size $m \times m$ and with a step size of 4 pixels. Then WAHOLB multi-resolution hierarchical features are extracted using Algorithm 1. Based on the Algorithm 1, for each patch, HOG and LBP descriptors are applied to obtain pixel-level features. Next, patch-level features are computed by concatenating HOG and LBP histograms, statistical characterization of wavelet sub-band coefficients and each patch's spatial position information.

As it can be seen from the Fig. 1, after feature extraction, k-means clustering is applied to construct visual vocabulary. It should be noted that the feature extraction procedure is done for each resolution and k-means clustering is applied for all the extracted features. In the next step, the features at the image level are represented using BoF model (Fig. 1). Therefore, the proposed BoF model represents images with their features at the pixel, patch, and image level.

Finally, the extracted features are used for training the proposed RBFNN classifier (mPSO–OSD).

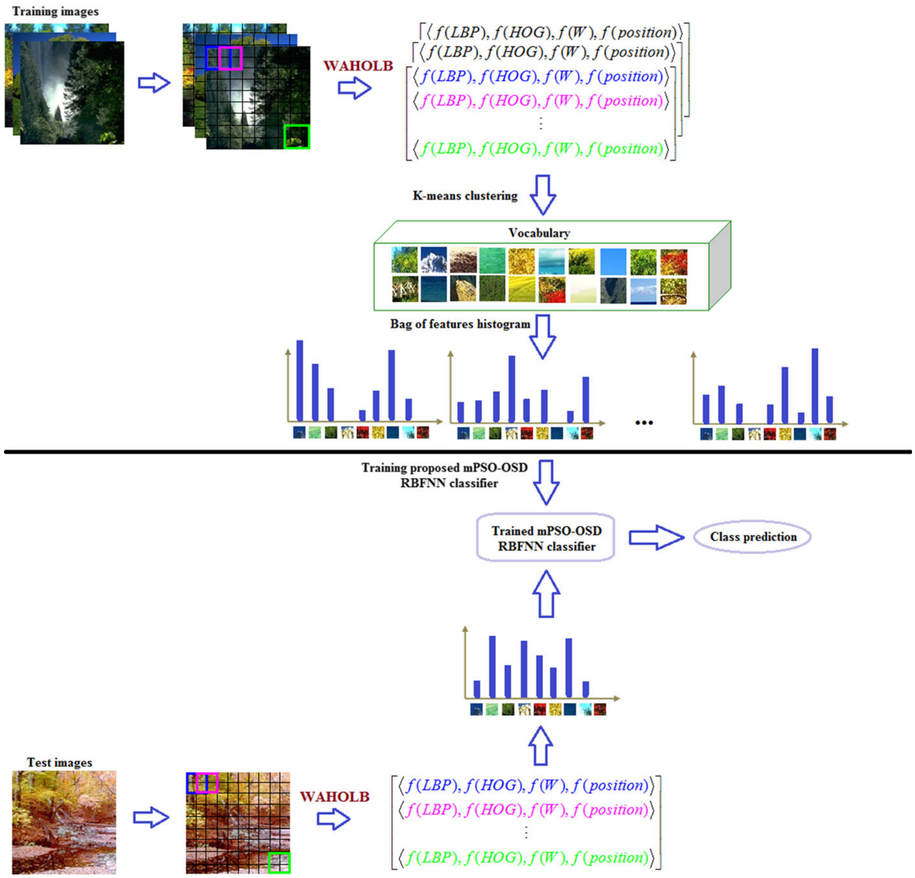

Fig. 1 Block diagram of the proposed model for constructing bag of visual words model

Algorithm 1. The pseudocode of the proposed WAHOLB descriptor

For each patch **Do**
Perform 2-level Haar wavelet decomposition
    Perform LBP coding
    Perform HOG coding
For each decomposition level **Do**
Compute LBP features
    Compute HOG features
    Compute Wavelet features
End For
    Compute LBP features of the entire patch
    Compute HOG features of the entire patch
    Compute features of the patch position
End For

2.1 Image Features

Different image features can be extracted for image representation. There are some reasons for integrating HOG, LBP, and wavelet coefficients for image representation. For an image with a simple background, HOG is able to represent its foreground accurately; however, its performance deteriorates for the images with complex or noisy backgrounds because HOG may consider some features extracted from the background as foreground features. As a complementary, LBP is robust against monotonic gray-scale changes and is able to filter out background noise through local binary texture patterns [36]. LBP groups non-uniform patterns (noise) together and separates them from the meaningful uniform patterns. Thus, noisy background does not significantly affect the similarity measure between images. Wavelet transform can provide a multi-resolution and orientation representations of images. In this article for each image, the L-level Haar wavelet decomposition is applied to obtain 4 subbands (i.e., approximation image and three detail coefficients in horizontal, vertical, diagonal orientations) at each level. In our implementation, we set $L=2$. The coefficients, obtained by low-pass filtering, correspond to the approximation of the image patch, and the detail coefficients contain high frequency components that resemble image details. Figure 2 shows the 2-level wavelet decomposition of an image in the forest category of the OT dataset.

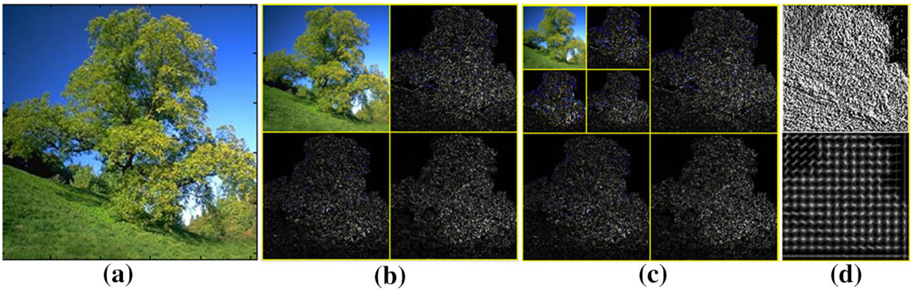

Fig. 2 This figure depicts 2-level Haar wavelet decomposition, HOG and LBP images. **a** Original image patch. **b** The first level decomposition, from *left* to *right* and *top* to *bottom*: approximation coefficients and three detail (*horizontal*, *vertical*, and *diagonal*) coefficients. **c** The second level decomposition, from *left* to *right* and *top* to *bottom*: approximation coefficients and three detail (*horizontal*, *vertical*, and *diagonal*) coefficients. **d** LBP image and HOG image from *top* to *bottom*

Based on the wavelet decomposition, LBP and HOG algorithms are applied to extract local pixel patterns on each original patch and its approximated coefficient matrices at each level (Fig. 2). Hence, multi-resolution pixel-level local features, i.e., wavelet subband coefficients, LBP and HOG features are obtained.

One of the key benefits of our method is that, it takes advantages of hidden information existed in different wavelet subbands in the horizontal, vertical and diagonal orientations. This produces highly efficient and compact image features for homogeneous areas.

LBP is computed as follows:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s\left(q_p - q_c\right) 2^p, \quad s\left(x\right) = \begin{cases} 1, \ x \geq 0 \\ 0, \ x < 0 \end{cases}, \tag{1}$$

where q_p and q_c denote the value of the central pixel (here the pixels are elements of an image patch or approximated coefficients) and its sampling neighbors, respectively, P is the number of neighbors, and R is the sampling radius in this neighborhood. The histogram of these binary numbers is then used to describe the texture of the image.

For the construction of the patch-based LBP, pattern histograms are built in cells. Then the histograms of the LBP patterns from different cells are concatenated to describe the texture of the current moving window. We use the notation $LBP^u_{P,R}$ to denote LBP feature that takes Psample points with radius R, and the number of 0–1 transitions is no more than u. A pattern that satisfies this constraint is called uniform pattern [36]. For instance, the pattern 0010010 is a non-uniform pattern for$LBP^2$and is a uniform pattern for LBP^4 because LBP^4 allows four 0–1 transitions. In this case, we get a feature vector of 58 bins instead of the original 256 bins. Moreover, the remaining non-uniform patterns are aggregated to a single bin, the uniform LBP histogram actually contains 59 bins. We use a 8×8 window around each pixel and compute the uniform pattern$LBP^{u_i}_{8,1}$ of each pixel. These descriptors are of 64-dimensional: $LBP_i = \left[LBP^{u_1}_{8,1}, LBP^{u_2}_{8,1}, \ldots, LBP^{u_{64}}_{8,1}\right]$.

As a dense version of the dominating SIFT feature, HOG features are extracted [4;32]. HOG has been widely accepted as one of the best features to capture the edge or local shape information. This descriptor gives the occurrences of gradient orientation in image patches [10].

To extract HOG features, each image is partitioned into 4×4 cells. Then for these cells a 1–9 histogram of gradient directions is stacked. Thus, a patch will be described by a $2 \times 2 \times 9 = 36$ dimensional feature vector (please see Fig. 3).

2.2 Patch-Level Feature Extraction

For constructing patch level features, image descriptors are densely applied to images. In addition to LBP and HOG features, statistical features extracted by wavelet transform and 1-D spatial information of each patch are extracted.

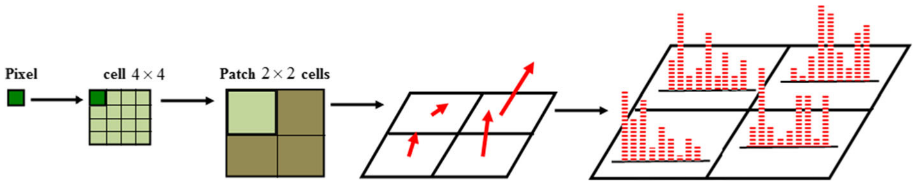

Fig. 3 Visualization of extracting HOG features

Thus, the patch-level features include: LBP features, $LBP_{8,1}^{u}$ histograms of the whole image patch and approximation coefficients, 64-dimensional HOG features, wavelet features: include the mean and the SD of magnitudes from each coefficient matrix and finally patch position information which measures the distance of center coordinates between each patch and the entire image.

The fusion is to directly concatenating the HOG descriptor, uniform LBP descriptor and wavelet features in the same patch as follows:

Step1 Extract a 36-dimensional HOG descriptor in each patch, denoted as HOG_i.

Step2 Choose a region of size 8×8 around the center of each patchand compute the uniform pattern $LBP_{8,1}^{u_i}$ of each pixel. These descriptors are composed as a 64-dimensional vector, i.e., $f(LBP_i) = \left[LBP_{8,1}^{u_1}, LBP_{8,1}^{u_2}, \ldots, LBP_{8,1}^{u_{64}}\right]$.

Step3 In each patch, HOG_i is fused with LBP_i and with the mean and SD of energies of detail coefficients as well as the 1-D spatial information.

Thus, a patch can be described as a compact feature vector in 141-dimensional space:

$$P_i = [f(LBP_i), f(HOG_i), f(W_i), f(position)].$$

3 The Proposed Radial Basis Function Neural Network Classifier

A RBFNN is a three-layer network. The layers include: input layer, hidden layer and output layer. The function of the input layer is to propagate input vectors to the hidden layer. The hidden layer includes a number of radial basis function units (n_h) with Gaussian kernels and bias (b_k). The j-th Gaussian function is determined by a center (c_j) and a width (σ_j). The RBFNN classifier starts by comparing the Euclidean distance between the input vector (x) and the center of the Gaussian function (c_j) and performs the nonlinear transformation in the hidden layer as follows:

$$h_j(x) = \exp\left(-||x - c_j||^2/\sigma_j^2\right), \tag{2}$$

where h_j is the notation for the output of the j-th neuron in the hidden layer. The linear operation of the output layer is given as follows:

$$y_k(x) = \sum_{j=1}^{n_h} w_{kj}.h_j(x) + b_k, \tag{3}$$

where y_k is the k-th output unit for the input vector x, w_{kj} is the weight connection between the k-th output unit and the j-th hidden layer unit, and b_k is the bias.

Considering Eqs. (2) and (3), it can be said that training of a RBFNN happens in three steps: (1) Determination of the number of the neurons in the hidden layer, (2) Determination of the centers and the widths of the Gaussian functions, and (3) Computing the synaptic weights between hidden and output layers.

The number of hidden neurons is determined by conducting different experiments. According to different applications of RBFNNs, many algorithms for determining the radial basis functions centers, widths and synaptic weights have been proposed in the literature. These proposals can be categorized into two major classes. The first class includes the algorithms in which the centers and the widths are not constant [11]. For instance, centers of the RBF units can be determined using k-means clustering technique or also by using self-organizing feature map method [11]. After center determination, widths of the RBF units usually are determined using p-nearest neighboring algorithm. The second class includes algorithms that assign constants values to the centers and the widths of the Gaussian functions [32].

Calculating the synaptic weights is the subject of the second class. For example, in [28], least-mean-square (LMS) is used for training the RBFNN, in [9] steepest decent (SD) is used for calculating synaptic weights. Pseudo-inverse (minimum-norm) method is employed in [7] for training RBFNN. In [32] authors propose quick propagation (QP) and optimum steepest decent (OSD) methods for training RBFNN. Several studies show that OSD leads high classification performance [18,32,33].

In [32], OSD was used to train the proposed RBFNN classifier. Random selection of the centers and the widths of the Gaussian functions were the major drawbacks of the work. To alleviate these shortcomings, Montazer et al. in [33] introduced three-phase learning method that used k-means and p-nearest neighbor (p-nn) algorithms to calculate the centers and the widths of the Gaussian functions, respectively. As a result, the performance of the proposed method in [32] was notably increased because of better initialization of the centers and the widths of the Gaussian functions. Even though the efficiency of the OSD was improved using the proposed method and also, reaching the global minimum in the weight space was met, it was probable that this method resulted in a suboptimal solution because the k-means algorithm was very sensitive to the initial choice of cluster centers and also, to noise. Accordingly, the proposed algorithm could get stuck in the local minima. Hence, Fathi and Montazer in [18] introduced a new learning method, called PSO–OSD, which outperformed the existing learning methods for the RBFNN classifier. In that work, center adjustment was done by the PSO clustering algorithm instead of the k-means clustering algorithm.

Despite the fact that the proposed clustering method is not sensitive to the initial values of the cluster centers, it is slow when the problem at hand is in high dimensional space. In addition, it may get stuck in the local minima due to the sprite of the proposed PSO algorithm.

To overcome these problems, an adaptive strategy is introduced to modify the PSO clustering algorithm in such a way that its convergence speed and global and local search abilities are highly improved.

3.1 Computing Centers of RBFNN Using the Modified PSO

In the PSO algorithm a particle i is defined by a position, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, a velocity, $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, a historical optimal position, $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and a historical optimal position of all particles which is denoted by $p_g = \left(p_{g1}, p_{g2}, \ldots, p_{gD}\right)$. Using the PSO algorithm, each particle searches in the solution space and updates its own velocity and position according to the following updating formulas [18]:

$$v_{id}^{(k+1)} = \omega v_{id}^{(k)} + c_1 r_1 \left(pbest_{id}^{(k)} - x_{id}^{(k)}\right) + c_2 r_2 \left(gbest_{d}^{(k)} - x_{id}^{(k)}\right), \tag{4}$$

$$x_{id}^{(k+1)} = x_{id}^{(k)} + v_{id}^{(k+1)}, \tag{5}$$

$$\omega = \omega_{max} - \frac{k}{k_{max}} \left(\omega_{max} - \omega_{min}\right), \tag{6}$$

where c_1 and c_2 are learning factors. These factors emphasize on the relative contribution of the overall best individual particles and the best direction of the particles; $r_1$and r_2 are two random numbers which are uniformly distributed in the interval [0, 1]; ω is inertia weight factor. It has notable influence on the performance of the PSO.

It is likely that PSO gets stuck into the local optimum and converges slowly in the late iterations. This drawback highly affects the overall performance of this optimization technique. Hence, we propose a novel adaptive strategy to tackle this difficulty. The proposed strategy dynamically adjusts the inertia weight factor ω, learning factors c_1 and c_2. This is achieved by considering the change of the search process.

If is large, a reliable global search is obtainable while the smaller value of increases the chance of a good local search. In the original PSO, the value of is set to 1 which causes a poor ability of a good search.

In [18] is decreased linearly using Eq. (6). This helps PSO to explore a larger area at the beginning and rapidly locates the approximate location of the optimal solution. As the value of decreases, particles slow down to start the local search. Although this method accelerates the convergence speed and improves the performance of the PSO algorithm, when the problem at hand is intricate and highly nonlinear, it does not provide a good situation for the PSO. Consequently, the PSO cannot provide good situation for strong global search in the late iterations. Thus, it fails to find the optimal solution.

To diminish these defects, we propose to modify the PSO algorithm as following:

$$v_{id}^{(k+1)} = \omega v_{id}^{(k)} + c_1^{(k)} r_1 \left(pbest_{id}^{(k)} - x_{id}^{(k)} \right) + c_2^{(k)} r_2 \left(gbest_{d}^{(k)} - x_{id}^{(k)} \right), \tag{7}$$

$$x_{id}^{(k+1)} = x_{id}^{(k)} + v_{id}^{(k+1)}, \tag{8}$$

$$\omega = \begin{cases} \omega - \frac{(\omega - \omega_{min})\left(\frac{1}{m}\sum_{j=1}^{m} f_j - f_i\right)}{f_{max} - \frac{1}{m}\sum_{j=1}^{m} f_i} & if \quad f_i < \frac{1}{m}\sum\limits_{i=1}^{m} f_i,\ m < n \\ \omega & if \quad \frac{1}{m}\sum\limits_{i=1}^{m} f_i \le f_i \le \frac{1}{n}\sum\limits_{i=1}^{n} f_i, \\ \omega_{max} - \frac{(\omega_{max} - \omega_{min})\left(f_i - \frac{1}{n}\sum_{i=1}^{n} f_i\right)}{f_{max} - \frac{1}{n}\sum_{i=1}^{n} f_i} & if \quad f_i > \frac{1}{n}\sum\limits_{i=1}^{n} f_i \end{cases} \tag{9}$$

$$\begin{cases} c_1^{(k)} = e^{-\left(\frac{k}{k_{max}}\right)^2} (c_{1min} - c_{1max}) + c_{1max} \\ c_2^{(k)} = e^{-\left(\frac{k}{k_{max}}\right)^2} (c_{2max} - c_{2min}) + c_{2min} \end{cases}, \tag{10}$$

where the $c_{1\max}$ and $c_{1\min}$ are the maximum and minimum values of c_1, respectively. The $c_{2\max}$ and the $c_{2\min}$ are the maximum and minimum values of c_2, respectively and the k_{max} is the maximum number of iterations. The ω_{max} and the ω_{min} are the maximum and minimum values of ω, respectively.The f_i is the value of fitness function of the current particle. The f_{max} is the maximum value of the fitness function in each iteration. The value of $\frac{1}{n}\sum_{i=1}^{n} f_i$ is the average value of the fitness function in each iteration and the value of $\frac{1}{m}\sum_{i=1}^{m} f_i$ is the average value of fitness function of m particles with fitness values smaller than the average value of the fitness function in each iteration.

In a high performance PSO algorithm, particles should have powerful global search ability in the early iterations and powerful local search ability in the late iterations. This is fulfilled by adjusting the value of ω using Eq. (9). This equation considers the fitness values for adjusting ω in such a way that for particles with good positions, inertia weigh ω is decreased in order to elude escaping from the global optimum. On the other hand, for particles with relatively good positions, inertia weigh ω is not updated as they possess good search ability. For some particles with larger fitness, inertia weigh ω is increased reasonably to better explore the search space. The proposed adaptive strategy ensures faster convergence speed of the PSO algorithm as well as it can efficiently manage the local and global search abilities.

In addition to ω, c_1 and c_2 play an important role in the performance of the PSO algorithm, c_1 is a self-learning factor and c_2 is a social learning factor, both of them are referred to set as 1.5 in the PSO algorithm used in [18].

At the very first iterations of the search process, strong self-learning ability and weak social learning ability are essential. This helps particles to explore the entire search space. On the contrary, in the late iterations, the PSO requires particles to have weak self-learning ability and strong social learning ability. This makes particles to find the global optimal solution. The proposed adaptive Eq. (10) satisfies the demanded criteria.

In order to comprehend the proposed clustering method using the modified PSO algorithm, suppose that each particle contains a solution, namely, it represents k cluster centroids vector which is denoted by $X = (M_1, M_2, ..., M_k)$ where $M_j = (s_{j1}, ..., s_{jl}, ..., s_{jf})$ refers to the j-th cluster centroid vector of a particle. The M_j is a column vector with f components that present feature vectors of the training data.

Now, the Euclidean distances between each training data and the cluster centers can be computed as following [18]:

$$d(M_{jl}, P_{rl}) = \sqrt{\sum_{l=1}^{f} (S_{jl} - t_{rl})^2} \quad for \quad 1 \leq j \leq k, 1 \leq r \leq n, 1 \leq l \leq f. \tag{11}$$

After all $d\left(M_{jl}, P_{rl}\right)$ are computed, feature l of pattern r is compared with the corresponding feature of cluster j, and then assigns 1 to $Z_{jrl} = 1$ where the Euclidean distance for each feature l of the pattern r is minimum [18]:

$$Z_{jrl} = \begin{cases} 1 & d(M_{jl}, P_{rl})\, is \text{ min} \\ 0 & \text{eslewhere} \end{cases}, \tag{12}$$

Therefore, the mean of the data, N_{jl}, for each solution can be calculated as follows [18]:

$$N_{jl} = \frac{\sum_{r=1}^{n} t_{rl} \times Z_{jrl}}{\sum_{r=1}^{n} Z_{jrl}} \; for \;\; 1 \leq j \leq k, 1 \leq l \leq f. \tag{13}$$

Hence, for each feature l of the cluster j, the Euclidean distances between mean of data N_{jl} and the centroid S_{jl} are calculated as follows [18]:

$$d(N_{jl}, S_{jl}) = \sqrt{\left(N_{jl} - S_{jl}\right)^2} \; for \;\; 1 \leq j \leq k, 1 \leq l \leq f. \tag{14}$$

Now, the fitness function for each cluster is obtained by summing the calculated distances as follows [18]:

$$F(M_j) = \sum_{l=1}^{f} d(N_{jl}, S_{jl}) \; for \;\; 1 \leq j \leq k. \tag{15}$$

Algorithm 2 shows the steps of the proposed method. Using this algorithm, we adjust RBF unit centers with *gbest* iterating for a certain number of iterations namely $k_{\max}$.

Algorithm 2. The pseudocode of the proposed PSO clustering for RBF unit center

```
For each Particle [i] Do
        Initialize Position vector X[i] in the range of maximum and minimum of dataset patterns
        Initialize Velocity vector V[i] in the range of [-a,a] (a = max(data)-min(data))
        Put initial Particle[i] into pbest_id[i]
Next i
While maximum iteration is not attained Do
        For each Particle [i] Do
                For each Cluster [j] Do
                        Compute the Fitness Function using Equations (13), (14) and (15)
                Next j
        Next i
        if run number is greater than 1 Then
                For each Particle [i] Do
                        For each Cluster [j] Do
                                if Fitness Function of Particle [i]'S Cluster [j] is better than Fitness Function of
                                pbest_id[i] 'S Cluster [j] Then
                                Put Cluster [j] of Particle [i] into Cluster [j] of pbest_id[i]
                                Endif
                        Next j
                Next i
        Endif
        For each Cluster [j] Do
                For each Particle [i] Do
                        Put the best of pbests in terms of Fitness Function into gbest
                Next i
        Next j
        Compute inertia weight ω using Equation (9)
        Compute c1 and c2 using Equation (10)
        For each Particle [i] Do
                Update Velocity vector V[i] using Equation (7)
                Update Position vector X[i] using Equation (8)
        Next i
Endwhile
Having processes algorithm, RBF unit centers are adjusted with gbest
```

For computing the widths of the RBFNN units, p-nearest neighboring algorithm is used.

3.2 Training the Proposed RBFNN

The OSD learning method is used to calculate the connection weights between hidden and the output layers of the network [32]. This method uses an optimum learning rate in each iteration of the training process. As stated in [32], the optimum delta weight vector (ODWV) can be determined as:

$$\Delta W_{opt} = \lambda_{opt} \Delta W = \frac{(E\phi)\,(E\phi)^T\,(E\phi)}{\left(E\phi\phi^T\right)(E\phi\phi^T)^T}, \tag{16}$$

hence

$$W_{new} = W_{old} + \frac{(E\phi)\,(E\phi)^T\,(E\phi)}{\left(E\phi\phi^T\right)(E\phi\phi^T)^T}, \tag{17}$$

which the initial value for W is chosen randomly.

4 Datasets and Experimental Results

In this section, at first, the results of modifying the PSO–OSD RBFNN is reported. Then the results of the proposed image classification method are discussed.

4.1 Experimental Results of Modifying the Three-Phased PSO–OSD RBFNN

The proposed classifier has been tested on popular datasets, namely, Proben1 dataset in the UCI machine learning repository [40] (Table 1 summarizes information about this dataset) and various synthetic and real life benchmark datasets.

The experiments have been done on a Windows 7 64-bit operating system, an Intel(R) Core(TM) i3-4130 CPU @ 3.40 GHz processor, and 16-GB RAM.

For training both classifiers, three-phased PSO–OSD (PSO–OSD) and modified PSO–OSD (mPSO–OSD), optimum steepest descent learning method has been employed to calculate the synaptic weights.

For Proben1, all experiments have been run 40 times and 50% of the datasets have been used as train set and the rest are used for validation and testing the network.

Tables 2, 3 present the operational parameters for training the PSO–OSD and mPSO–OSD optimizers.

For comparing the PSO–OSD and the mPSO–OSD, precision, classification error and training time of 40 runs are reported in Table 4.

On Iris dataset precision has been increased from 92 to 97.1%, on Wine dataset precision has been increased from 75.5 to 84.3%. As well as for Abalone, Ionosphere and Glass datasets precisions have been increased by 7.52, 5 and 6.7%, respectively. Therefore, for Proben1 on average the precision has been increased about 6.26%. This considerable improvement shows that the mPSO–OSD outperforms the PSO–OSD in terms of classification performance and training time.

Table 1 Description of the five-benchmark problem

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 PSO–OSD

Parameters	Description	Considered value
ω_{max}	Maximum inertia weigh	1.1
ω_{min}	Minimum inertia weigh	0.4
C_1	Local search coefficient	1.5
C_2	Social search coefficient	1.5
Population size	Number of particles	15
k_{max}	Number of iterations	400
Number of runs	–	40

Table 3 Description and values of the parameters used in modified PSO–OSD

Parameters	Description	Considered value
ω_{max}	Maximum inertia weigh	1.1
ω_{min}	Minimum inertia weigh	0.4
C_1	Local search coefficient	[3,58]
C_2	Social search coefficient	[3,58]
Population size	Number of particles	15
k_{max}	Number of iterations	400
Number of runs	–	40

The reason that mPSO–OSD is faster than PSO–OSD is because of the proposed adaptive strategy that avoids PSO from getting trapped in local minimum and also provides strong local and global search abilities so that fast speed and high performance of the mPSO–OSD classifier are guaranteed compared to the PSO–OSD classifier.

Moreover, Table 4 shows the SD of the classification results for the mPSO–OSD and the PSO–OSD. The mPSO–OSD obtains smaller values, which means that it possesses better repeatability for the retrain process than PSO–OSD. Additionally, on all datasets, the mPSO–OSD decreases the standard deviation of the classification error for the test sets.

We also applied mPSO–OSD to synthetic and real life benchmark datasets. The real life benchmark datasets can be downloaded from the UCI machine learning repository [2], except from the gas furnace dataset, which can be found in [6]. A summary of these datasets can be found in Table 5.

- *Ailerons–elevators* These datasets contain information about an F16 airplane. The aim is to predict the control action on the ailerons and the elevators of the aircraft. The condition of the airplane is used as features.
- *Auto MPG* By using this dataset, we can measure the fuel consumption of a car.
- *Auto price* This dataset is used for prediction of the price of an automobile based on different features of the automobile.
- *Boston housing* Boston Housing dataset is used for predicting the prices of houses in Boston based on some specifications of the houses.
- *CPU small* CPU small dataset includes a set of computer-system activity measures from a Sun-SPARC station. This dataset is used for predicting the portion of time that the CPUs run in user mode.
- *Friedman* This dataset is sampled from the following function [20]:

  $$y = 5[2\sin(\pi x_1 x_2) + 4\left(x_3 - 0.5\right)^2 + 2x_4 + x_5\right] + \varepsilon,$$

  where ε is Gaussian noise $\sim N(0, 0.8)$. Values for the input variables are sampled from a uniform distribution in [0, 1].
- *Gas furnace* This dataset includes prediction of the CO_2 concentration y(t) in the exhaust gases of a gas furnace system, while changing the methane feed rate u(t). The optimum integration of input variables results in the following model [39]:

  $$y(t) = RBF\left(u(t-2), y(t-2), y(t-1)\right).$$

- *House price-16H* The aim of this dataset is to predict the average price of houses based on demographic composition and the state of the housing market.
- *Machine CPU* The goal is to predict the relative CPU performance using as inputs CPU characteristics.

Table 4 Classification error (%) and precision (%) for the mPSO–OSD and the PSO–OSD

Data set	mPSO–OSD						PSO–OSD					
	Training set				Testing set		Training set				Testing set	
	Ave.	SD	Ave.	SD	Precision	Time (s)	Ave.	SD	Ave.	SD	Precision	Time (s)
Iris	2.67	1.63	3.73	1.98	97.1	6.13	4.51	3.42	5.37	3.37	92	8.90
Wine	24.49	2.04	28.76	2.97	84.3	5.76	11.64	2.8	32.46	4.61	75.5	10.45
Abalone	41.21	0.52	46.72	0.53	69.92	186.90	42.83	0.72	47.02	0.74	62.4	254.02
Ionosphere	7.77	2.04	14.66	2.36	94.5	13.51	28.34	2.42	42.5	3.54	89.5	19.91
Glass	19.79	3.10	35.57	5.18	82.8	8.53	31.21	3.83	42.09	7.35	76.1	12.37

Table 5 Benchmark datasets

Dataset	No. of input	No. of examples
Real life datasets		
Ailerons	40	13,750
Auto MPG	6	392
Auto Price	15	159
Boston housing	13	506
CPU small	12	8192
Elevators	18	16,599
Gas furnace	3	290
House price-16H	16	22,784
Machine CPU	6	209
Synthetic datasets		
Friedman	5	1000
Mackey–Glass	6	1995
Samad	3	1500

- *Mackey–Glass* This dataset is a chaotic time series. It is widely used for measuring the performance of different neural networks [23,29,45]; in the discrete case it is generated from the following formula:

$$y\left(t+1\right)=\left(1-b\right)y\left(t\right)+a\frac{y\left(t-\tau\right)}{\left(1+y^{10}\left(t-\tau\right)\right)}.$$

  We try to predict the current value of the time series $y\left(t\right)$, based on the previous six values:

$$y\left(t\right)=RBF\left(y\left(t-1\right),y\left(t-2\right),\ldots,y\left(t-6\right)\right),$$

  The parameter values a = 0.2, b = 0.1, and $\tau = 30$ have been used for generating the dataset. Gaussian noise $\sim N(0, 0.05)$ was added to the output $y\left(t\right)$ and also for letting system reach to its steady state the first 1000 values are discarded.
- *Samad* This dataset is generated from the function [44]:

$$y=\frac{1}{1+exp[-exp\left(x_1\right)+\left(x_2-0.5\right)^2+\sin\left(\pi x_3\right)]}+\varepsilon,$$

where ε is Gaussian noise $\sim N(0,0.025)$. Values for the input variables are sampled from a uniform distribution in [0, 1].

Except for the gas furnace and Mackey–Glass datasets, we randomly divided each dataset to training, validation and testing sets (50% testing, 25% validation and 25% testing sets).

For comparison purposes we compared our proposed method with RBFNN proposed by Alexandridis et al. [1].

Alexandridis et al. introduced a new training algorithm for radial basis function networks [1]. In that paper a new framework for training RBF networks is proposed using a non-symmetric fuzzy partitioning of the input space (NS) and a novel optimization technique using PSO for optimizing the functionality of fuzzy means (FM). The proposed training algorithm (NSFM) used FM to select the number of RBF centers and their locations while linear regression method was used to compute the synaptic weights.

In order to achieve stable and comparable results with PSO–NSFM, we run our proposed algorithm 30 times on each dataset. These experiments have been repeated with 30 different random selections so the average and SD results can be evaluated and compared with the results reported in [1].

It can be seen from Table 6 that, in all the cases except for Auto Price and Boston Housing the mPSO–OSD defeats the PSO–NSFM. By noticing to these two datasets, we see that for Auto Price dataset, there are 15 features and 159 samples. In addition, for Boston Housing dataset, there are 13 features and 506 samples. Compared to the other datasets the number of samples is relatively small. This shows that compared to the PSO–NSFM, the mPSO–OSD needs more training samples in order to achieve high performance.

By comparing the average RMSE values found by the two methods, we can say that the values obtained by mPSO–OSD are smaller than the values obtained by the PSO–NSFM. Considering the SD values of RMSE, we see that they are relatively small; this demonstrates that the mPSO–OSD achieves results that are more consistent and the training process of this network is repeatable.

The experiments show that in facing with high dimensional data and a relatively large amount of samples, the proposed mPSO–OSD reaches very promising results. This fact illustrates that the proposed mPSO–OSD outstandingly copes with highly nonlinear and complex datasets.

Table 6 also shows the training time of the both neural networks. Although the implementation environments of the two methods are different, due to the simpler structure of the mPSO–OSD as well as its high global and local search abilities, compared to the PSO–NSFM, it is faster.

4.2 Experimental Results of Scene Classification

In this section, we compare our proposed method with baseline methods such as SIFT, PHOG, Gradient LBP, GaborLBP, GIST and other state-of-the-art methods. We implemented our proposed method on the following datasets.

Oliva and Torralba (OT) dataset Oliva and Torralba [37] dataset includes 2688 images classified as 8 categories: 360 coasts, 328 forest, 260 highway, 308 inside of cities, 374 mountain, 410 open country, 292 streets and 356 tall buildings. Note that river and forest scenes are all considered as forest. Moreover, there is not a specific sky scene since almost all of the images contain the sky object.

These annotations make a higher inter-class variability. Most of the scenes present a large intra-class variability. The average size of each image is 250×250 pixels. The class numbers in our experiments are: ‘coast = 1’, ‘forest = 2’, ‘highway = 3’, ‘inside city = 4’, ‘mountain = 5’, ‘open country = 6’, ‘street = 7’ and ‘tall building = 8’.

Fei-Fei and Perona (FP) dataset Fei-Fei and Perona [19] dataset contains 13 categories and is only available in grayscale. This dataset consists of the 2688 images (8 categories) of the OT dataset plus: 241 suburb residence, 174 bedroom, 151 kitchen, 289 living room and 216 office. Note that the presence of man-made images (e.g. bedroom, living room etc.) allows a low inter-class variability since these images have a similar structure. The average size of each image is approximately 250×300 pixels.

The class numbers in our experiments are: ‘suburb = 1’, ‘coast = 2’, ‘forest = 3’, ‘highway = 4’, ‘inside city = 5’, ‘mountain = 6’, ‘open country = 7’, ‘street = 8’, ‘tall building = 9’, ‘office = 10’, ‘bedroom = 11’, ‘kitchen = 12’, ‘living room = 13’.

Table 6 Results comparisons of modified PSO–OSD (mPSO–OSD) and PSO–NSFM

Dataset	Method	RMSE validation	RMSE testing	Number of nodes	Time (s)
Ailerons	mPSO–OSD	1.20×10^{-4}	1.09×10^{-4}	217	1105
	PSO–NSFM	1.57×10^{-4}	1.56×10^{-4}	237	4220
Auto MPG	mPSO–OSD	1.63	1.27	27	24
	PSO–NSFM	2.11	1.95	36	92
Auto price	mPSO–OSD	2.38×10^{3}	2.27×10^{3}	39	116
	PSO–NSFM	2.17×10^{3}	2.11×10^{3}	57	339
Boston housing	mPSO–OSD	2.83	2.34	87	867
	PSO–NSFM	2.35	2.72	138	2588
CPU S-mall	mPSO–OSD	1.96	1.83	284	1163
	PSO–NSFM	2.80	2.81	467	4505
Elevators	mPSO–OSD	1.46×10^{-3}	1.53×10^{-3}	816	1928
	PSO–NSFM	2.08×10^{-3}	2.11×10^{-3}	981	7197
Friedman	mPSO–OSD	10.43×10^{-1}	1.5	92	31
	PSO–NSFM	9.08×10^{-1}	1.06	114	133
Gas furnace	mPSO–OSD	1.38×10^{-1}	2.79×10^{-1}	18	19
	PSO–NSFM	1.70×10^{-1}	4.27×10^{-1}	22	77
House price-16H	mPSO–OSD	2.34×10^{4}	2.27×10^{4}	783	3249
	PSO–NSFM	3.62×10^{4}	3.61×10^{4}	984	13,201
Machine CPU	mPSO–OSD	1.21×10^{1}	2.09×10^{1}	15	21
	PSO–NSFM	2.28×10^{1}	2.16×10^{1}	19	83
Mackey–Glass	mPSO–OSD	2.36×10^{-2}	2.61×10^{-2}	17	127
	PSO–NSFM	4.79×10^{-2}	5.30×10^{-2}	24	539
Samad	mPSO–OSD	2.06×10^{-2}	2.18×10^{-2}	54	13
	PSO–NSFM	2.47×10^{-2}	2.80×10^{-2}	89	60

Lazebnik, Schmid and Ponce (LSP) dataset Lazebnik et al. [27] dataset contains 15 categories and, as with FP, is only available in grayscale. This dataset consists of the 13 categories of the FP dataset plus: 315 store and 311 industrial. The average size of each image is approximately 250 × 300 pixels. The class numbers in our experiments are: 'suburb = 1', 'coast = 2','forest = 3', 'highway = 4', 'inside city = 5', 'mountain = 6', 'open country = 7', 'street = 8', 'tall building = 9', 'office = 10', 'bedroom = 11', 'industrial = 12', 'kitchen = 13', 'living room = 14' and 'store = 15'.

4.2.1 Experimental Setup

We extracted 141-dimensional WAHOLB descriptors and 128-dimensional SIFT descriptors from 32 × 32 overlapping patches. 3400-dimensional PHOG descriptors were extracted for 4 spatial pyramids in the interval $[0, 2\pi]$ and with 40 gradient orientation bins [5]. 512-dimensional GIST descriptors were extracted from images divided into 4 × 4 blocks in 8 orientations and 4 scales [37]. For extracting GaborLBP [62] and GradientLBP [24],

Table 7 Parameter selection of the proposed PSO–OSD RBFNN scene classification datasets

Dataset	No. of train	No. of test	No. of hidden neurons	No. of epochs	Size of the vocabulary
OT	806	1882	285	60,000	1200
FP	1351	2508	550	2,750,00	1500
LSP	1570	2250	600	2,900,000	1500

Table 8 Performance comparison on OT dataset

Method	Mean average precision (mAP%)
WAHOLB	95.2
PHOG	76.4
GIST	80.4
GaborLBP	79.9
GradientLBP	73.3
SIFT	86.6
Proposed method in [37]	89
Proposed method in [3]	91.49
Proposed method in [63]	91.1
Proposed method in [54]	90.2
Proposed method in [41]	90.30 ± 2.54
Proposed method in [42]	92.49 ± 1.99
Proposed method in [65]	89.2
Proposed method in [31]	88.10 ±0.90
Proposed method in [56]	91.71

the magnitude image was divided into 10 × 10 non-overlapping patches and then $LBP^{u}_{8,1}$ histograms over all regions were extracted and fused. GaborLBP was extracted using Gabor filters in 5 scales and 8 orientations. Therefore, for GaborLBP image descriptors are in 40,000-dimensional and for GradientLBP in which the LBP operators ($LBP^{u}_{8,1}$ and $LBP^{u}_{16,2}$) have been applied to the original image and gradient magnitude image obtained by Sobel operator, the final image descriptors are of 5600-dimensional.

Table 7 shows the parameters used in the mPSO–OSD RBFNN classifier and the size of the constructed vocabularies.

4.2.2 Experimental Results

We use linear support vector machines [12] for classifying randomly selected training images (with a fixed number of images per class) and then evaluate the classification performance on the rest of the images. We performed 10-fold cross-validation to determine hyperparameters on each randomly selected training set and reported the test accuracy averaged over 10 trials.

Tables 8, 9, 10 show the results of comparison between WAHOLB, SIFT, PHOG, GIST, GaborLBP, GradientLBP and other state-of-the-arts methods on OT, FP and LSP datasets, respectively.

Table 9 Performance comparison on FP dataset

Method	Mean average precision (mAP%)
WAHOLB	89.7
PHOG	71.2
GIST	76.1
GaborLBP	72.3
GradientLBP	68.9
SIFT	83
Proposed method in [19]	65.2
Proposed method in [27]	74.7
Proposed method in [63]	85.9 ± 1.3
Proposed method in [41]	87.63 ± 2.3
Proposed method in [65]	86.1

Table 10 Performance comparison on LSP dataset

Method	Mean average precision (mAP%)
WAHOLB	89.1
PHOG	63.5
GIST	70.6
GaborLBP	70.8
GradientLBP	65.4
SIFT	80.1
Proposed method in [27]	81.4 ± 0.5
Proposed method in [35]	76.67 ± 0.39
Proposed method in [3]	85.62%
Proposed method in [63]	83.7 ± 1.3
Proposed method in [54]	82.91
Proposed method in [41]	85.16 ± 1.62
Proposed method in [42]	87.39 ± 2.48
Proposed method in [65]	84.2
Proposed method in [31]	84.10 ± 0.96

Based on the reported results, we see that cosidering the overall scene classification accuracy, the proposed WAHOLB descriptor outperforms all the other image descriptors, i.e., SIFT, PHOG, GIST, GaborLBP, GradientLBP and other state-of-the-art methods. It emphasizes that WAHOLB based multi-resolution features are strong enough for scence classification task.

WAHOLB based descriptor outperforms SIFT on the three datasets by rather a good difference (at least 6.7%, at most 11%). This is because of the advantages of wavelet features and LBP in the texture analysis.

An interesting observation is that the LBP based magnitude coding plays an important role in extracting local structural information. These local patterns and the wavelet sub-band information with dense spatial sampling make our proposed descriptor to be robust

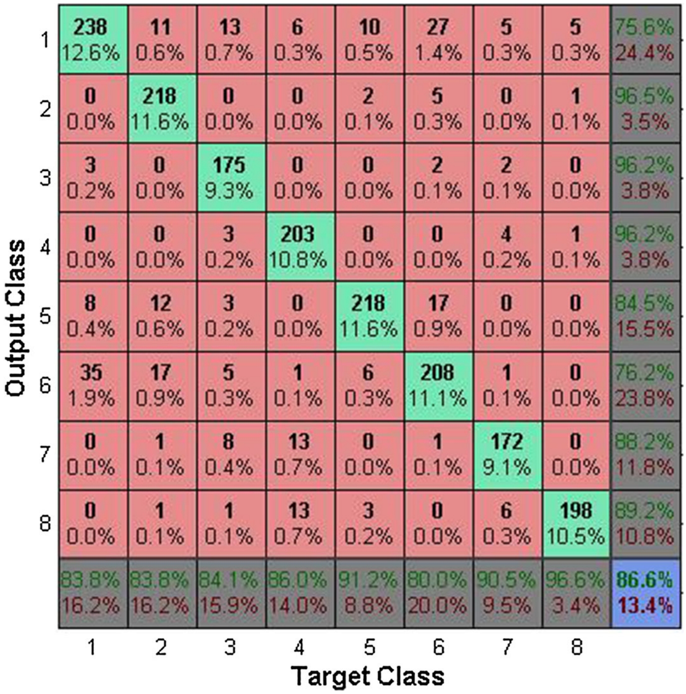

Fig. 4 Confusion matrix resulted by SIFT (OT dataset)

and discriminant. In contrast to our proposed method, PHOG is based on the global spatial layout, this makes it unsuitable and often biased in feature pooling when significant appearance of a scene changes or heavy illumination changes occur. This fact emphasizes that the perceptual properties of the multi-resolution features, which are used in WAHOLB, contain rich cues to semantic category, and they are informative and discriminant for visual classification.

GaborLBP defeats PHOG, GIST and GradientLBP. It shows the discriminating power of combining multiscale and multi-orientation Gabor filters with LBP.

Figures 4, 5 show the confusion matrices of the SIFT method and the proposed method, respectively. These figures state that, using WAHOLB descriptor the performance of the scene

classification increases nearly 8.6% on OT dataset. This improvement happens for all of the classes by 9.9, 9.1, 9.2, 8.6, 5.9, 14.2, 6.3 and 3.4%, respectively.

The confusion matrix in Fig. 4 demonstrates that SIFT method confuses among coast, mountain and open country categories. This is because of the fact that a large area of the images in these categories is composed of the same textural regions so SIFT fails to discriminate these categories. On the other hand, confusion matrix in Fig. 5 relates that by using WAHOLB descriptor this confusion is alleviated.

Another important observation that should be mentioned is the role of the mPSO–OSD RBFNN classifier. We trained and tested linear SVM with our proposed descriptor and we reached 91.1, 86.3 and 85.6% accuracies for OT, FP and LSP datasets, respectively. This shows that the proposed classifier has notably affected the performance of the classification by 4.1, 3.4 and 3.5% on OT, FP and LSP datasets, respectively.

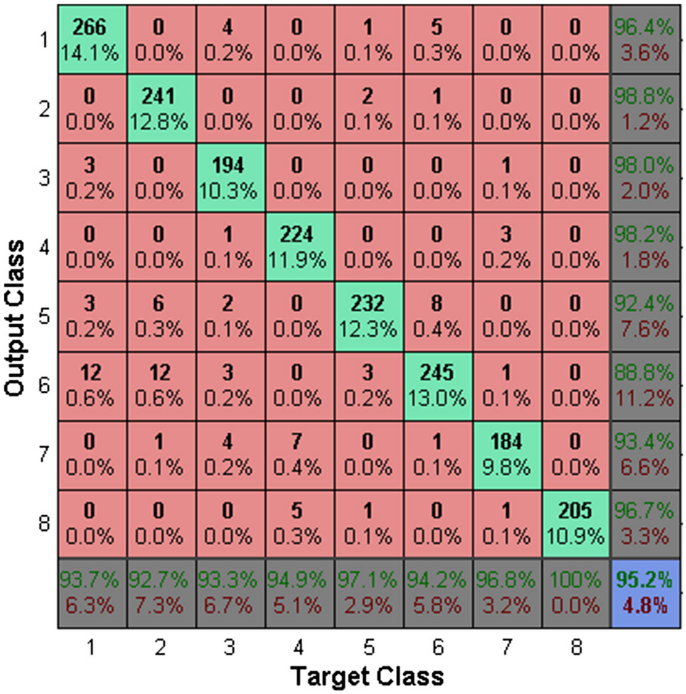

Fig. 5 Confusion matrix resulted by our proposed method (OT dataset)

5 Conclusion

This article proposes a high performance scene classification approach including a new image descriptor, so-called WAHOLB, and introducing a new variant of RBFNN classifier. WAHOLB is based on wavelet theory; HOG and LBP feature extraction methods. It extracts multi-resolution hierarchical features from local image patterns from the pixel, patch, and image levels. This allows WAHOLB to better represent image semantics. As a second contribution, we modify PSO algorithm so that its local and global search abilities have been improved so that is can converge faster. The modified PSO algorithm is then used for adjusting the center values of Gaussians in RBFNN classifier. Experimental results demonstrated the superiority of our proposed scene classification approach on OT, FP, and LSP benchmark datasets.

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

References

1. Alexandridis A, Chondrodima E, Sarimveis H (2013) Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization. IEEE Trans Neural Netw Learn 24(2):219–230. doi:10.1109/Tnnls.2012.2227794

2. Asuncion A, Newman DJ (2007) UCI machine learning repository. University of California, Irvine. http://archive.ics.uci.edu/ml/
3. Bolovinou A, Pratikakis I, Perantonis S (2013) Bag of spatio-visual words for context inference in scene classification. Pattern Recognit 46(3):1039–1053. doi:10.1016/j.patcog.2012.07.024
4. Bosch A, Zisserman A, Munoz X (2006) Scene classification via pLSA. In: Proceedings of computer vision-Eccv 2006, Pt 4, 3954: 517–530
5. Bosch A, Zisserman A, Munoz X (2007) Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, pp 401–408
6. Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. Wiley, Hoboken
7. Cancelliere R, Gai M (2003) A comparative analysis of neural network performances in astronomical imaging. Appl Numer Math 45(1):87–98. doi:10.1016/S0168-9274(02)00237-4
8. Chan T, Jia K, Gao S, Lu J, Zeng Z, Ma YP (2014) A simple deep learning baseline for image classification? arXiv preprint. arXiv preprint arXiv:1404.3606 1(3)
9. Chen XY (2007) Deformation measurement of the large flexible surface by improved RBFNN algorithm and BPNN algorithm. In: Proceedings advances in neural networks-ISNN 2007, Pt 3, 4493: 41–48
10. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of CVPR IEEE, pp 886–893
11. Dong C-R, Chan PP, Ng WW, Yeung DS (2011) A survey of the initialization of centers and widths in radial basis function network for classification. In: 2011 IEEE international conference on machine learning and cybernetics (ICMLC), pp 1082–1087
12. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J (2008) LIBLINEAR: a library for large linear classification. J Mach Learn Res 9:1871–1874
13. Fan H, Zhou E (2016) Approaching human level facial landmark localization by deep learning. Image Vis Comput 47:27–35
14. Farhidzadeh H, Zhou M, Goldgof DB, Hall LO, Raghavan M, Gatenby RA (2014) Prediction of treatment response and metastatic disease in soft tissue sarcoma. Med Imaging Comput Aided Diagn. doi:10.1117/12.2043792
15. Farhidzadeh H, Chaudhury B, Zhou M, Goldgof DB, Hall LO, Gatenby RA, Gillies RJ, Raghavan M (2015) Prediction of treatment outcome in soft tissue sarcoma based on radiologically defined habitats. Proc Spie. doi:10.1117/12.2082324
16. Farhidzadeh H, Goldgof DB, Hall LO, Gatenby RA, Gillies RJ, Raghavan M (2015) Texture feature analysis to predict metastatic and necrotic soft tissue sarcomas. IEEE Syst Man Cybern. doi:10.1109/Smc.2015.488
17. Farhidzadeh H, Kim JY, Scott JG, Goldgof DB, Hall LO, Harrison LB (2016) Classification of progression free survivalwith nasopharyngeal carcinoma tumors. In: SPIE medical imaging, international society for optics and photonics
18. Fathi V, Montazer GA (2013) An improvement in RBF learning algorithm based on PSO for real time applications. Neurocomputing 111:169–176. doi:10.1016/j.neucom.2012.12.024
19. Fei-Fei L, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories 2005. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, 2: 524–531
20. Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19:1–67
21. Heikkila M, Pietikainen M, Schmid C (2009) Description of interest regions with local binary patterns. Pattern Recognit 42(3):425–436. doi:10.1016/j.patcog.2008.08.014
22. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
23. Holmes CC, Mallick BK (2000) Bayesian wavelet networks for nonparametric regression. IEEE Trans Neural Netw 11(1):27–35. doi:10.1109/72.822507
24. Huang X, Li SZ, Wang Y (2004) Shape localization based on statistical method using extended local binary pattern. In: Third international conference on IEEE image and graphics (ICIG'04), pp 184–187
25. Jain AK, Farrokhnia F (1990) Unsupervised texture segmentation using gabor filters. In: 1990 IEEE international conference on systems, man, and cybernetics: 14–19. doi:10.1109/Icsmc.1990.142050
26. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp 1097–1105
27. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). pp 2169–2178
28. Lin IC, Liou CY (2007) Least-mean-square training of cluster-weighted modeling. In: Proceedings artificial neural networks-ICANN, Pt 2, 4669: 301–310

29. Loo CK, Rajeswari M, Rao MVC (2004) Novel direct and self-regulating approaches to determine optimum growing multi-experts network structure. IEEE Trans Neural Netw 15(6):1378–1395. doi:10.1109/Tnn.2004.837779
30. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110. doi:10.1023/B:Visi.0000029664.99615.94
31. Meng X, Wang Z, Wu L (2012) Building global image features for scene recognition. Pattern Recognit 45(1):373–380
32. Montazer GA, Sabzevari R, Khatir HG (2007) Improvement of learning algorithms for RBF neural networks in a helicopter sound identification system. Neurocomputing 71(1–3):167–173. doi:10.1016/j.neucom.2007.08.002
33. Montazer GA, Sabzevari R, Ghorbani F (2009) Three-phase strategy for the OSD learning method in RBF neural networks. Neurocomputing 72(7–9):1797–1802. doi:10.1016/j.neucom.2008.05.011
34. Montazer GA, Soltanshahi MA, Giveki D (2015) Extended bag of visual words for face detection. Adv Comput Intell 9094:503–510. doi:10.1007/978-3-319-19258-1_41 (Pt I Iwann 2015)
35. Montazer GA, Giveki D (2015) Content based image retrieval system using clustered scale invariant feature transforms. Opt Int J Light Electron Opt 126(18):1695–1699
36. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal 24(7):971–987. doi:10.1109/Tpami.2002.1017623
37. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175. doi:10.1023/A:1011139631724
38. Pang YW, Yan H, Yuan Y, Wang KQ (2012) Robust CoHOG feature extraction in human-centered image/video management system. IEEE Trans Syst Man Cybern B 42(2):458–468. doi:10.1109/Tsmcb.2011.2167750
39. Patrinos P, Alexandridis A, Ninos K, Sarimveis H (2010) Variable selection in nonlinear modeling based on RBF networks and evolutionary computation. Int J Neural Syst 20(05):365–379
40. Prechelt L (1994) Proben1: A set of neural network benchmark problems and benchmarking rules
41. Qin J, Yung NH (2010) Scene categorization via contextual visual words. Pattern Recognit 43(5):1874–1888
42. Qin J, Yung NH (2012) Feature fusion within local region using localized maximum-margin learning for scene categorization. Pattern Recognit 45(4):1671–1683
43. Quelhas P, Monay F, Odobez JM, Gatica-Perez D, Tuytelaars T, Van Gool L (2005) Modeling scenes with local descriptors and latent aspects. In: IEEE international conference on computer vision: 883–890
44. Samad T (1991) Back propagation with expected source values. Neural Netw 4(5):615–618
45. Shi ZW, Han M (2007) Support vector echo-state machine for chaotic time-series prediction. IEEE Trans Neural Netw 18(2):359–372. doi:10.1109/Tnn.2006.885113
46. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
47. Song TC, Li HL (2013) Local polar DCT features for image description. IEEE Signal Proc Lett 20(1):59–62. doi:10.1109/Lsp.2012.2229273
48. Sun S (2013) A survey of multi-view machine learning. Neural Comput Appl 23(7–8):2031–2038
49. Tian XL, Jiao LC, Liu XL, Zhang XH (2014) Feature integration of EODH and color-SIFT: application to image retrieval based on codebook. Signal Process Image 29(4):530–545. doi:10.1016/j.image.2014.01.010
50. Tian Y, Luo P, Wang X, Tang X (2015) Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision. pp 1904–1912
51. Walia E, Pal A (2014) Fusion framework for effective color image retrieval. J Vis Commun Image Represent 25(6):1335–1348. doi:10.1016/j.jvcir.2014.05.005
52. Wang XY, Han TX, Yan SC (2009) An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th international conference on computer vision (ICCV), pp 32–39. doi:10.1109/Iccv.2009.5459207
53. Wang W, Yang X, Ooi BC, Zhang D, Zhuang Y (2016) Effective deep learning-based multi-modal retrieval. VLDB J 25(1):79–101
54. Wang Y, Gong S (2007) Conditional random field for natural scene categorization. In: BMVC. Citeseer, pp 1–10
55. Wang R, Tao D (2016) Non-local auto-encoder with collaborative stabilization for image restoration. IEEE Trans Image Process 25(5):2117–2129
56. Wang S, Wang Y, Zhu S-C (2012) Hierarchical space tiling for scene modeling. In: Asian conference on computer vision. Springer, pp 796–810

57. Wang N, Yeung D-Y (2013) Learning a deep compact image representation for visual tracking. In: Advances in neural information processing systems. pp 809–817
58. Wu JX, Rehg JM (2011) CENTRIST: a visual descriptor for scene categorization. IEEE Trans Pattern Anal 33(8):1489–1501. doi:10.1109/Tpami.2010.224
59. Xu C, Tao D, Xu C (2013) A survey on multi-view learning. arXiv preprint arXiv:1304.5634
60. Yu J, Qin ZC, Wan T, Zhang X (2013) Feature integration analysis of bag-of-features model for image retrieval. Neurocomputing 120:355–364. doi:10.1016/j.neucom.2012.08.061
61. Yu J, Rui Y, Tang YY, Tao D (2014) High-order distance-based multiview stochastic learning in image classification. IEEE Trans Cybern 44(12):2431–2442
62. Zhang WC, Shan SG, Gao W, Chen XL, Zhang HM (2005) Local gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. IEEE Int Conf Comput Vis 1:786–791
63. Zhang S, Tian Q, Hua G, Huang Q, Gao W (2014) ObjectPatchNet: towards scalable and semantic image annotation and retrieval. Comput Vis Image Underst 118:16–29
64. Zheng YB, Huang XS, Feng SJ (2010) An image matching algorithm based on combination of SIFT and the rotation invariant LBP. J Comput Aided Design Comput Gr 22(2):286–292
65. Zhou L, Zhou Z, Hu D (2013) Scene classification using a multi-resolution bag-of-features model. Pattern Recognit 46(1):424–433
66. Zhu Z, Wang X, Bai S, Yao C, Bai X (2016) Deep learning representation using autoencoder for 3d shape retrieval. Neurocomputing 204:41–50