

Image Content Analysis Using Modular RBF Neural Network

Chuan-Yu Chang^{1, *}, Hung-Jen Wang², and Chi-Fang Li¹

¹ Department of Computer Science and Information Engineering

National Yunlin University of Science & Technology

Yunlin 640, Taiwan, ROC

{chuanyu, g9417707}@yuntech.edu.tw

² Graduate School of Engineering Science and Technology

National Yunlin University of Science & Technology

Yunlin 640, Taiwan, ROC

g9210810@yuntech.edu.tw

Received 2 May 2010; Revised 12 June 2010; Accepted 30 June 2010

Abstract. Image content analysis has become an important issue in multimedia processing. Region-based image retrieval systems attempt to reduce the gap between high-level semantics and low-level features by representing images at the object level. Recently, the radial basis function (RBF) neural network has been proposed to solve the classification problem; however, it is time-consuming and sensitive to center initialization. Therefore, modular RBF neural network (MRBFNN) incorporated with a self-organizing map (SOM) and a learning vector quantization (LVQ) neural network is proposed for semantic-based image content classification. Using SOM and LVQ, we can obtain more appropriate centers for the RBF neural network. Moreover, principal component analysis (PCA) is applied to reduce the dimension of features. Experimental results show that the proposed method is capable of analyzing components of photographs into semantic categories with high accuracy, resulting in photographic analysis that is similar to human perception.

Keywords: Image content analysis, region-based image retrieval, PCA, SOM, RBF

1 Introduction

Content-based image retrieval is an important topic in computer vision and multimedia computing due to the rapid development in digital imaging storage and network technologies. For large databases with thousands of images, effective indexing is an important issue in content-based image retrieval. Various content-based image retrieval systems (CBIR) have been proposed, such as QBIC [1], Virage [2], VisualSeek [3], MARS [4], and SIMPLiCity [5]. These systems represent an image using a set of feature attributes such as color, texture, shape, and layout. These features are stored with the image in the database. A retrieval is performed by matching the feature attributes of the query image with the features stored in the database. However, users typically do not think in low-level features; i.e., user queries use semantics (e.g., “show me a sunrise image”). Most image retrieval systems have poor performance for semantics queries [6-9]. Thus, how to group images into semantically meaningful categories or index images in a database based on the low-level visual features of the images is a problem.

A successful categorization of images greatly improves the performance of content-based image retrieval systems by filtering out irrelevant images during the matching stage. Many proposed schemes use semantic labeling approaches [6, 7, 10, 11, 12]. However, an image may contain many objects and it is difficult to recognize a large number of objects in an image. Region-based retrieval systems (RBIR) attempt to overcome the gap between high-level semantics and low-level features by representing images at the object level [7, 8, 13]. A region-based retrieval system applies the image segmentation method to decompose an image into several regions, which may contain a complete object if the segmentation is perfect. The object representation is thus close to human visual perception. Classification is a very important task for image content analysis. A good and efficient feature extraction method should guarantee that the classification accuracy can be improved greatly. For instance, in [14], the principal component analysis (PCA) method was used to extract linear principal components (PCs) for multispectral images. The classification accuracy of PC images was improved significantly compared to that of a method that directly used the original multispectral data. However, image segmentation is almost as difficult as image understanding because the images are 2D projections of 3D objects. The Netra [15] and

* Correspondence author

Blobworld [16] systems compare images based on individual regions. Part of the comparison task is shifted to the users. To query an image, users provide the segmented regions to be matched and the attributes of the region such as color and texture to be used for evaluating similarity. This query scheme increases the burden on users.

Although, region-based systems aim to decompose images into constituted objects, each segmented region is indirectly related to its semantics. There is no clear mapping from a set of region images to semantics; therefore, most systems are confined to matching images with low-level features. In order to improve retrieval results, high-level concept-based indexing must be considered. However, it is very difficult to bridge the semantic gap between low-level image features and high-level concepts.

In recent years, neural network methods have been proposed to solve the classification problem [17-21]. The radial basis function neural network (RBFNN) is the most popular architecture because it has good learning and approximation capacity. However, RBFNNs are sensitive to center initialization; the centers of an RBFNN need to be appropriately set. There exists several strategies such as random selection, systematic seeding, expert contribution, clustering, editing methods, which been usually used for selecting the right number of hidden neurons. Among them, the clustering methods, like k-means, fuzzy c-means or their variants, are the most common strategies. However, they are often not suitable for RBFNN owing to they are unsupervised techniques of classification. They are able to reduce the number of hidden neurons, but they cannot determine their appropriate number. Actually, the supervised techniques work in a more satisfactory way. Decision trees, quantization methods, and other initialization approaches, such as condensing techniques and genetic algorithms are widely used [22]. In addition, the training procedure of a traditional RBFNN is time-consuming, especially when the dimension of the inputs is high. Moreover, the low-level features may contain redundant information.

To overcome these drawbacks, a high-level semantic category should be mapped into low-level features using the proposed MRBF neural network (MRBFNN). We also adopt principal component analysis (PCA) to reduce the feature dimension and to derive effective and discriminative features [14, 22, 23]. In addition, the HSV color space, statistical texture analysis, and shape features are used to represent the segmented objects because these features are closely related to human visual perception. We use the MRBFNN to improve the classification accuracy and to speed up the training time. A combination of SOM and LVQ neural networks is used to select appropriate centers for RBFNNs.

Our proposed system consists of training and testing stages. Fig. 1. shows a flowchart of the proposed semantic-based image analysis system. In the training stage, the training images are segmented into several regions using J-image segmentation (JSEG) [24]. Next, the segmented regions are manually classified into semantic categories by the user. Since JSEG segmentation results usually exist over segmentation phonomoment, a user interaction process is provided to merge regions into semantic objects. Low-level image features such as color, texture, and shape are extracted from each category and then incorporated into the proposed modular RBF neural network (MRBFNN) after the feature dimension is reduced by PCA. In addition, in the MRBFNN, a combination of SOM and LVQ is applied to overcome the problem of center initialization in a traditional RBFNN. In the testing stage, the test image is segmented into regions by JSEG. Then the low-level image features are extracted from each region. PCA is applied to sift significant features and to reduce the feature dimension. Finally, the sifted features from each segmented region are fed into the trained MRBFNN for semantic analysis. We use a percentage to describe the semantic category contained in the test image.

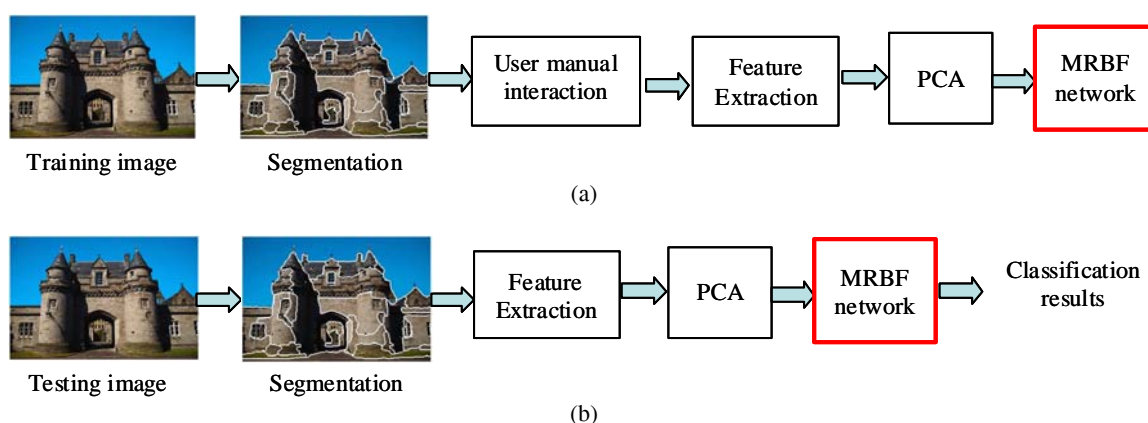


Fig. 1. Flowchart of semantic-based photographic analysis. (a) Training stage and (b) Testing stage

The proposed semantic-based photograph analysis system has the following advantages: (1) the semantic category of an image is estimated using low-level features extracted from the whole image and object features such as color, texture, and shape; (2) a reasonable accuracy rate is achieved after a rough image segmentation using JSEG without employing user interaction to assign semantic labels; (3) the problem of center initialization in a traditional RBFNN is overcome by the combination of SOM and LVQ neural networks; and (4) the pro-

posed MRBF neural network has a faster learning rate and a higher accuracy rate for real world images than those of a traditional RBF network.

The rest of this paper is organized as follows: In Section II, the methods of segmentation and feature extraction, the principle component analysis (PCA), the self-organizing map (SOM) neural network, and the learning vector quantization (LVQ) neural network are briefly introduced. The modular RBF neural network is presented in Section III. Experimental results are shown in Section IV, and the conclusion is given in Section V.

2 Related Work

2.1 Image Segmentation

Real world images contain many object regions with various colors and textures. In order to connect low-level features obtained from object regions to high-level semantics, image segmentation has to be performed first. In this paper, we use JSEG, a state-of-the-art segmentation technique for color image segmentation [25], to segment real world images into several regions. The basic process of the JSEG method consists of two stages: color space quantization and spatial segmentation. In the first step, JSEG quantizes colors of an image to several representative classes, and then labels pixels with the color classes to form a class map of the image. Then, the image is segmented using multi-scale J-images on the class map, which can be viewed as a spatial type of texture composition. Although JSEG segmentation is reasonable, the segmented results may not be intuitive for human perception. In addition, JSEG can only segment a color image into several regions; it cannot semantically label it. Therefore, a user interaction process is provided for users to manually assign a semantic category label for each segmented region. Fig. 2(a) shows an image of a building. Fig. 2(b) and (c) show the JSEG segmentation results and user category labeling, respectively.

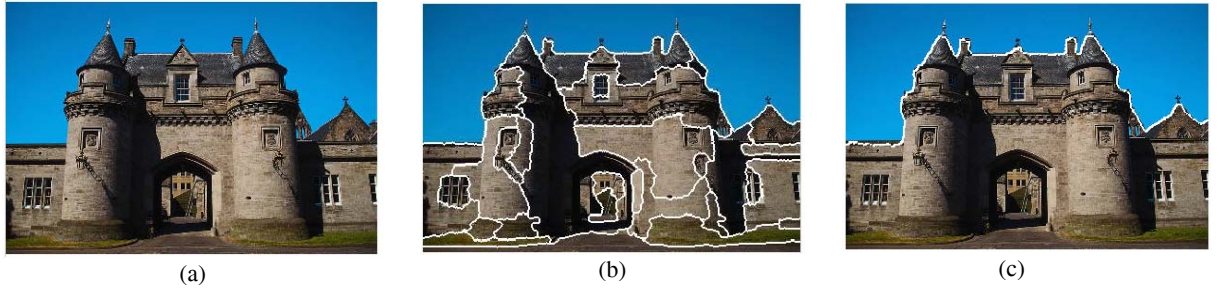


Fig. 2. (a) Original image, (b) Segmented image, and (c) Manually assigned categories

2.2 Feature Extraction

After a user has manually assigned a semantic category label for each segmented region, each region is described by its color, texture, and shape features. The HSV (hue, saturation, value) color histogram is quantified into 64 color bins to represent the color features of an object region. Fig. 3(d) and 3(e) shows the HSV features of the building and sky objects, respectively. To describe the texture features of an object region, five common gray-level statistical features and with four angles (0, 45, 90, and 135) are used, as shown in Eq. (1-5) [26-28].

$$\text{Contrast} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 P_{\delta}(i, j), \quad (1)$$

$$\text{Energy} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{\delta}(i, j)^2, \quad (2)$$

$$\text{Entropy} = -\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{\delta}(i, j) \log(P_{\delta}(i, j)), \quad (3)$$

$$\text{Homogeneity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P_{\delta}(i, j)}{1 + (i-j)^2}, \quad (4)$$

$$\text{Max probability} = \text{Max}[P_{\delta}(i, j)], \tag{5}$$

where $P_{\delta}(i, j)$ is the co-occurrence matrix under specific conditions $\delta(r, \theta)$, where r denotes the distance and θ is the orientation between two adjacent intensities (i, j) . Because the edge direction histogram [29] has the characteristics of shift invariance and scale invariance, it is used to describe the shape information. The algorithm for generating an edge direction histogram has three stages: detecting edges, finding important edges, and quantizing edge orientation. The Sobel operator is used for edge detection to obtain the gradient image. It generates two edge components, G_x and G_y . The amplitude and edge orientation are then computed using Eq. (6-7), respectively. Next, the important edges of the gradient image are extracted by comparing the edges to a threshold value T_I , which is chosen to be 25. Finally, the edge histogram is uniformly quantized into n segments $\angle G_1, \angle G_2, \dots, \angle G_n$. 24 histogram bins are used in our system. The results of the edge direction histogram are shown in Fig. 3(f-g). Each image can be represented as a feature vector that consists of color, texture, and shape features, with a feature vector dimension size of 108.

$$|G| = \sqrt{G_x^2 + G_y^2}, \tag{6}$$

$$\angle G = \tan^{-1}(G_x / G_y). \tag{7}$$

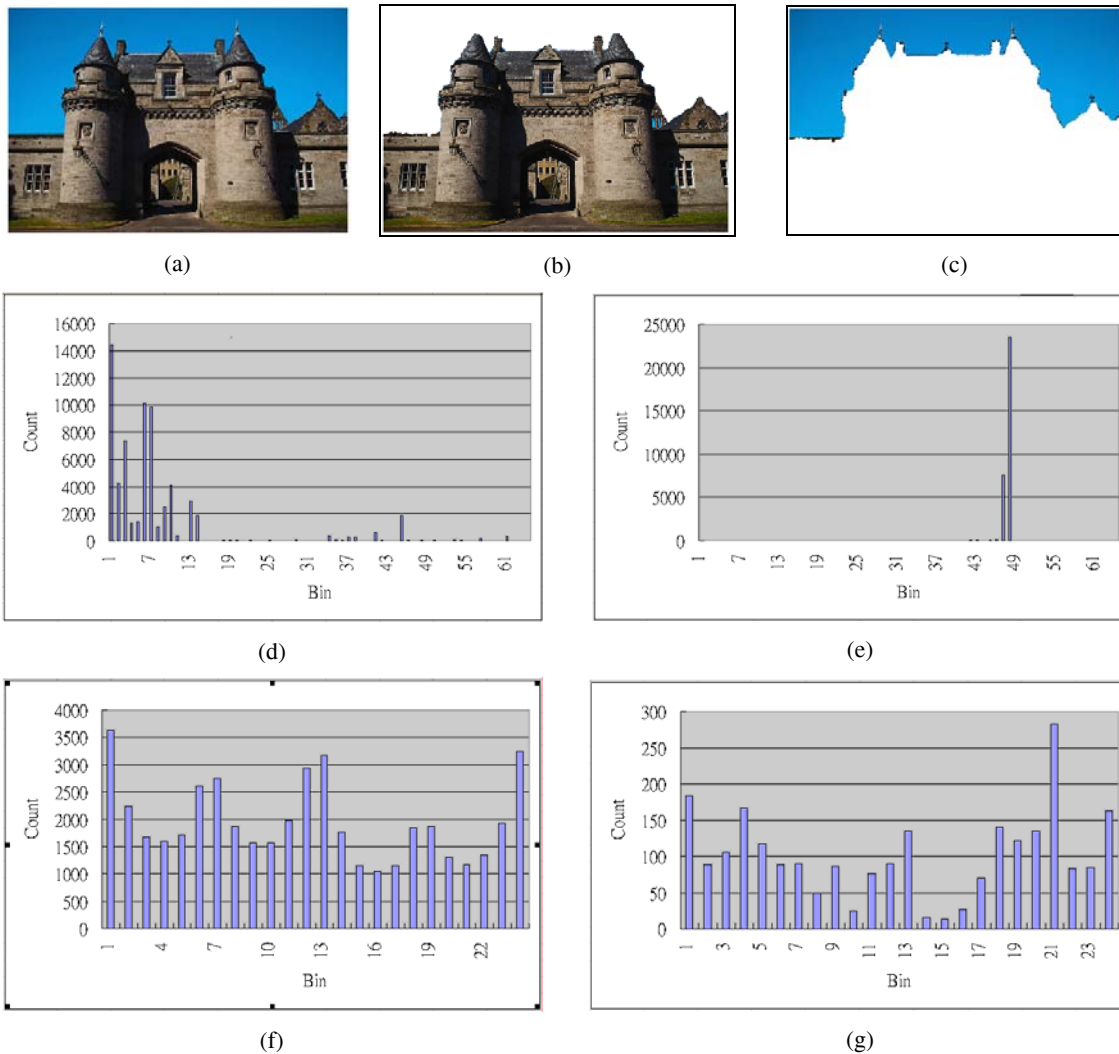


Fig. 3. (a) Original image, (b) Segmented building image, (c) Segmented sky image, (d) HSV color histogram of the segmented building, (e) HSV color histogram of the segmented sky, (f) Edge histogram of the building region quantized into 24 bins, (g) Edge histogram of the sky region quantized into 24 bins

2.3 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a useful statistical technique that has been applied in various fields such as image compression and dimensional reduction [4, 23, 24]. PCA can be used to reduce a large set of variables to a small set while retaining most of the information of the large set. The generated principal components are uncorrelated, and they have the maximal variance.

Suppose that N image features are extracted from each image to form an original feature vector $Y = \{y_1, y_2, \dots, y_N\}$. The symmetric covariance matrix, C_y , of the original feature vector is calculated as:

$$C_y = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})(y_i - \bar{y})^T, \quad (8)$$

where \bar{y} denotes the mean of Y .

Subsequently, the eigenvalues of the symmetric covariance matrix are calculated and ordered in a decreasing sequence as $\lambda_1, \lambda_2, \dots, \lambda_N$. The corresponding eigenvectors, E_1, E_2, \dots, E_N , respectively, can then be found. One can create an ordered orthogonal basis with the first eigenvector that has the direction of the largest variance of the data. This gives the components in the order of significance. The k -th principal component, x_k , is given by :

$$x_k = E_k^T Y \quad k = 1, 2, 3, \dots, N. \quad (9)$$

To obtain enough principal components, we compute the energy preservation factor P , by retaining only M eigenvalues:

$$P = \frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^N \lambda_i} \text{ where } N > M. \quad (10)$$

Using PCA, we obtain a new feature set $\{x_1, x_2, \dots, x_M\}$. In addition to the dimensional reduction, this set of feature variables can also have the maximal variance. For this reason, each segmented region is represented by a lower dimensional feature vector, which is an input of the MRBF neural network.

2.4 Combining SOM and LVQ Neural Networks to Generate Appropriate Centers for RBF Neural Network

RBF neural networks are sensitive to center initialization, so selecting significant features for center initialization is important. The SOM neural network has been proven to have the capacity of autoclustering. In our proposed scheme, the SOM neural network is applied to obtain appropriate centers for MRBF neural network initialization. The self-organizing map developed by Kohonen is an unsupervised, competitive learning, clustering neural network, in which only one neuron is “fired” at a time. Therefore, the neural network can achieve an automatic formation of topologically correct maps of features of observable events. In other words, SOM transforms input patterns into one-dimensional or two-dimensional maps of features in a topologically ordered fashion. Fig. 4. shows the architecture of the self-organizing neural network, which is a fully connected network.

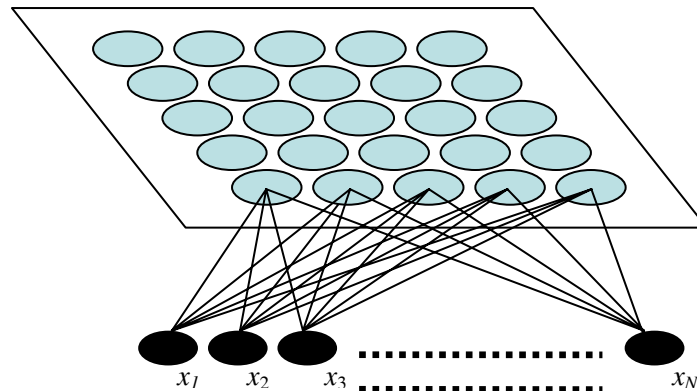


Fig. 4. Architecture of a self-organizing neural network

The formation of the self-organizing map involves three essential processes:

1. Competition process
2. Synaptic Adaptation process
3. Cooperation process

2.4.1 Competition Process

The input of the neural network can be written in vector form as:

$$X = [x_1, x_2, \dots, x_N]. \quad (11)$$

The synaptic weight vector of neuron k in the two-dimensional array is given as:

$$v_k = [v_{k1}, v_{k2}, \dots, v_{kN}], \quad k = 1, 2, \dots, J. \quad (12)$$

The best match of the input vector X with the synaptic weight vector v_k is determined using:

$$q(X) = \min_{v_k} \|X - v_k\|_2, \quad k = 1, 2, \dots, J, \quad (13)$$

where $q(k)$ is the index into the output neuron array that specifically identifies the winning neuron, and $\|\cdot\|_2$ is the Euclidean norm.

2.4.2 Synaptic Adaptive Process and Cooperative Process

After the winning neuron is identified, the synaptic weight vector associated with the winning neuron and the neurons within a defined neighborhood of the winning neuron is given by:

$$v_k(t+1) = v_k(t) + \mu(t)h_{k,q}(t)[X(t) - v_k(t)], \quad (14)$$

where the learning rate parameter varies with time. It starts at an initial value μ_0 , and then decreases gradually with increasing time t . This is defined as:

$$\mu(t) = \mu_0 \exp\left(-\frac{t}{\tau_0}\right), \quad (15)$$

where τ_0 is the time constant. $h_{k,q}(t)$ is the neighborhood function centered around the winning neuron $q(k)$ at the discrete time index t . It is defined as:

$$h_{k,q}(t) = \exp\left(-\frac{d_{k,q}^2}{2\sigma^2(t)}\right), \quad (16)$$

where the lateral distance $d_{k,q}$ and $\sigma(t)$ are respectively defined as:

$$d_{k,q}^2 = \|r_k - r_q\|^2, \quad (17)$$

and

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right), \quad (18)$$

r_k and r_q represent the coordinates of neurons k and q , respectively. σ_0 is the initialized value and τ_1 is a time constant. As time t increases, the width $\sigma(t)$ decreases at an exponential rate. Accordingly, the size of the topological neighborhood shrinks with time. In practice, both neighborhood $h_{k,q}(t)$ and learning rate parameter μ_t are relatively wide/large at the beginning of training and then shrink monotonically with time.

The algorithm of the proposed SOM is summarized as follows:

Step 1. Initialization:

Randomly assign the initial synaptic weights. Initialize the learning rate parameter. Define the topological neighborhood function.

Step 2. Competition process:

Use Eq.(13) to find the best-matching neuron $q(k)$ at time step t .

Step 3. Adaptive Process:

Adjust the synaptic weight vectors of all neurons using the update formulas in Eq. (14).

Step 4. Cooperative Process:

Update the learning rate and neighborhood function using Eqs.(15-18).

Step 5. Repeat Step 2 through Step 4 until the weight adjustments become negligibly small.

To obtain the appropriate initial centers, the inputs of SOM are the feature vectors extracted by PCA. Suppose we have N regions which can obtain N feature vectors using PCA calculated as the input vector of the SOM neural network; we can then generate weight vectors $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_J\}$. Once the SOM algorithm has been trained, the feature map computed by the algorithm shows important statistical characteristics of the input space. The SOM algorithm provides an approximate method for computing the clusters in an unsupervised manner, with the approximation being specified by the synaptic weight vectors of the neurons in the feature map. The computation of the feature map may be viewed as the first of two stages for adaptively solving a pattern classification problem, as shown in Fig. 5. The second stage involves learning vector quantization (LVQ), which provides a mechanism for the final fine tuning of a feature map. LVQ is a supervised learning technique that uses class information to move the clusters slightly to improve the quality of the classifier decision regions. An input vector is randomly picked from the input space. If the class labels of the input vector and a cluster agree, the cluster is moved in the direction of the input vector. On the other hand, if the class labels of the input vector and the cluster disagree, the cluster is moved away from the input vector. The anti-reinforced learning rule of the learning vector quantization (LVQ) algorithm is adopted to move similar clusters slightly. Let $\{\mathbf{v}_j\}_{j=1}^J$ denote the weight vector of the SOM neural network, and $\{\mathbf{x}_i\}_{i=1}^N$ denote the feature vector of the training patterns. For each input feature vector $\mathbf{x}_i(n)$, the best-matching weight vector of the SOM neural network is identified by the condition:

$$k = \arg \min_j \{\|\mathbf{x}_i - \mathbf{v}_j\|\}. \quad (19)$$

Let $\ell_{\mathbf{v}_k}$ denote the class associated with the weight vector of the SOM neural network, and $\ell_{\mathbf{x}_i}$ denote the class label of the input vector \mathbf{x}_i . The weight vector of the SOM neural network is adjusted as follows:

If $\ell_{\mathbf{v}_k} = \ell_{\mathbf{x}_i}$, then

$$\mathbf{v}_k(n+1) = \mathbf{v}_k(n) + \eta(n)[\mathbf{x}_i - \mathbf{v}_k(n)]. \quad (20)$$

If $\ell_{\mathbf{v}_k} \neq \ell_{\mathbf{x}_i}$, then

$$\mathbf{v}_k(n+1) = \mathbf{v}_k(n) - \eta(n)[\mathbf{x}_i - \mathbf{v}_k(n)], \quad (21)$$

where the learning rate $\eta(n)$ decreases gradually during iterations n , $0 \leq \eta(n) \leq 1$.

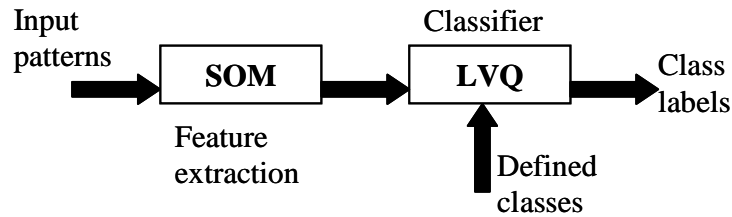


Fig. 5. Adaptive pattern classification system using SOM and LVQ

3 MRBF Neural Networks

The proposed MRBF combines SOM and LVQ networks to generate an appropriate center of initiation. In addition, the MRBF neural network (MRBFNN) adopts the concept of modular processing, which is applied in hidden layers for dimension transformation. Since modular processing is used, each output neuron is calculated using the sum of linear modules and is not inter-related. This design enables each RBF module to work independently, which speeds up the network training time and improves the classification capacity of individual RBF modules. In addition, the modular processing can easily add (or renew), delete, or change any subnetwork

of the RBF neural network (RBFNN) without affecting successfully trained modules. Fig. 6. shows the architecture of the MRBF neural network, which consists of M hidden modules and M output neurons. Since the MRBFNN is a supervised neural network, the category number needs to be defined before the MRBF is trained. There are M RBFNNs in the MRBF, with each semantic category is defined in one RBFNN. Therefore, only M semantic categories are classified; each region segmented by JSEG is classified into one of the M semantic categories. The parametric model and learning capability of the MRBFNN are similar to those of the RBFNN. The parametric model of the MRBFNN can be given as:

$$f_j(\mathbf{x}) = \sum_{i=1}^k w_{ji} \phi_{ji}(\|\mathbf{x} - \mathbf{c}_{ji}\|), \quad \forall j = 1, 2, \dots, M, \quad (22)$$

where $\mathbf{x} \in R^N$ is an input vector using PCA, ϕ_{ji} is the basis function of the MRBFNN, w_{ji} is the weight in the output layer, $\mathbf{c}_{ji} = (c_{ji1}, c_{ji2}, \dots, c_{jin})^T$ denotes the i -th center of the j -th module obtained by SOM and LVQ neural networks, and $\|\cdot\|$ denotes the Euclidean norm.

If the basis function of the MRBFNN is a Gaussian function, then Eq.(12) can be written as:

$$f_j(\mathbf{x}) = \sum_{i=1}^k w_{ji} \phi_{ji}(\|\mathbf{x} - \mathbf{c}_{ji}\|) = \sum_{i=1}^k w_{ji} \exp(-\|\mathbf{x} - \mathbf{c}_{ji}\|^2 / \sigma_{ji}^2), \quad (23)$$

where σ_{ji} is the i -th bandwidth of the Gaussian function of the j -th module, defined as:

$$\sigma_{ji} = \frac{d_{max}}{\sqrt{k}}, \quad (24)$$

where d_{max} is the maximum Euclidean distance between the selected center of the j -th module, and k is the center number of the j -th module.

The MRBFNN consists of various RBFNNs. Therefore, the error cost function of the MRBFNN is defined as:

$$J(n) = \frac{1}{2} \sum_{j=1}^M \left[f_j^{desired}(n) - \sum_{i=1}^k w_{ji}(n) \exp(-\|\mathbf{x}(n) - \mathbf{c}_{ji}(n)\|^2 / \sigma_{ji}^2(n)) \right]^2, \quad (25)$$

where $f_j^{desired}$ denotes the desired output of the j -th output neuron and $f_j^{desired} = \begin{cases} 1, & j\text{-th categories} \\ 0, & \text{otherwise} \end{cases}$.

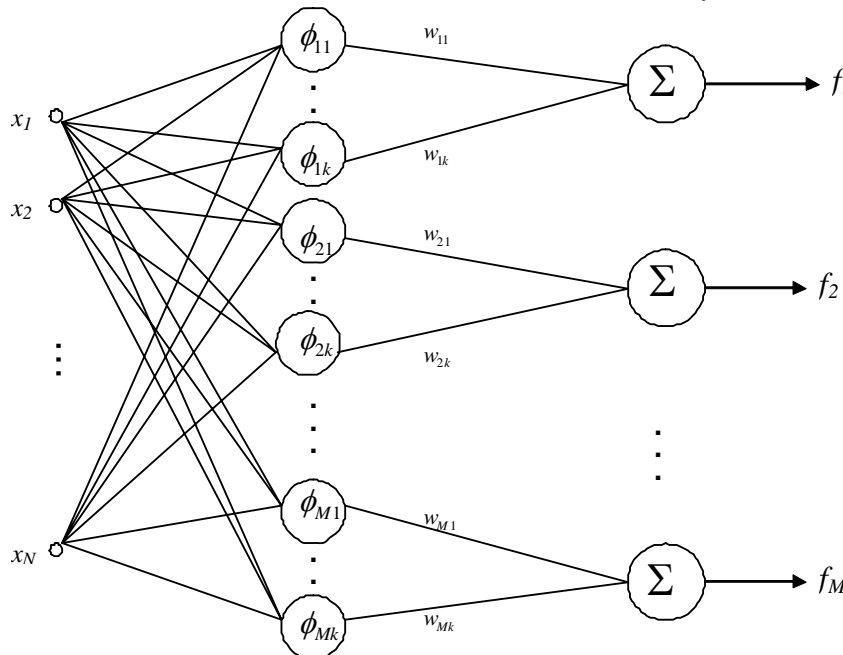


Fig. 6. MRBF neural network architecture

In the MRBFNN, we adopt the least mean squares (LMS) algorithm to define the updated equations for the MRBFNN parameters. They are given as:

$$w_{ji}(n+1) = w_{ji}(n) - \mu_w \frac{\partial}{\partial w_{ji}} J(n) |_{w_{ji}=w_{ji}(n)} = w_{ji}(n) + \mu_w e(n) \psi(n), \quad (26)$$

$$\begin{aligned} \mathbf{c}_{ji}(n+1) &= \mathbf{c}_{ji}(n) - \mu_c \frac{\partial}{\partial \mathbf{c}_{ji}} J(n) |_{\mathbf{c}_{ji}=\mathbf{c}_{ji}(n)} \\ &= \mathbf{c}_{ji}(n) + \mu_c \frac{e(n)w_{ji}(n)}{\sigma_{ji}^2(n)} \exp(-\|\mathbf{x}(n) - \mathbf{c}_{ji}(n)\|^2 / \sigma_{ji}^2(n)) [\mathbf{x}(n) - \mathbf{c}_{ji}(n)], \end{aligned} \quad (27)$$

$$\begin{aligned} \sigma_{ji}(n+1) &= \sigma_{ji}(n) - \mu_\sigma \frac{\partial}{\partial \sigma_{ji}} J(n) |_{\sigma_{ji}=\sigma_{ji}(n)} \\ &= \sigma_{ji}(n) + \mu_\sigma \frac{e(n)w_{ji}(n)}{\sigma_{ji}^2(n)} \exp(-\|\mathbf{x}(n) - \mathbf{c}_{ji}(n)\|^2 / \sigma_{ji}^2(n)) \|\mathbf{x}(n) - \mathbf{c}_{ji}(n)\|^2, \end{aligned} \quad (28)$$

$$\text{where } \psi(n) = [\phi_{j1}(\|\mathbf{x}(n) - \mathbf{c}_{j1}(n)\|), \dots, \phi_{jN}(\|\mathbf{x}(n) - \mathbf{c}_{jN}(n)\|)]^T, \quad (29)$$

$$e(n) = f_j^{desired}(n) - f_j(n), \quad (30)$$

and μ_w , μ_c and μ_σ are appropriate learning rate parameters.

The MRBFNN training steps are as follows.

Step 1. Use the PCA method to reduce the input feature dimension.

Step 2. Choose the centers for the MRBFNN using the SOM neural network.

Step 3. Adjust the centers of the SOM neural network using the LVQ algorithm.

Step 4. Calculate the initial value of the spread parameter for the MRBFNN using Eq. (24).

Step 5. Initialize the weights in the output layer of the MRBFNN to random values.

Step 6. Provide the input vector of the j -th class and compute the neural network output using Eq. (22).

Step 7. Update the neural network parameters using Eqs. (26-28).

Step 8. Stop if the neural network has converged; otherwise, go back to Step 6.

When the MRBFNN converges, the segmented regions of test images are fed into the trained MRBFNN for classification. The category of each region is determined by $q = \arg \max_{\forall m} f_m$, $m = 1, 2, \dots, M$, where f_m is the output of the m -th RBFNN.

4 Experimental Results and Discussions

There are about 3000 scene images in the Corel image collection. A total of 247 real-world images (150 dpi) were selected from the Corel image collection for training. The rest of the scene images were used for testing. The experiment environment for performance evaluation was implemented using Borland C++ Builder 6 on an ASUS PC equipped with an Intel Pentium-IV 2.8GHz CPU and 1GB of RAM. Sky and water were assigned to the same semantic category because their features are very similar. According to [5-7, 10, 30-32], there are eight common categories in natural scenes images: Building, Cloud, Sky or Water, Grass, Tree, Flower, Mountain, and Sunrise. The number of images of each category used for training and the representational color of each category are shown in Table 1. In general, each real-world image may contain several object regions. In order to increase the accuracy of the content analysis, an image must be correctly segmented before training. Each real-world image was first segmented into several regions using the JSEG segmentation algorithm. Although JSEG segmentation is reasonable, no semantic labels are produced. Accordingly, we manually assigned a semantic category label to each object region as mentioned in Section 2.1. Next, the low-level features, which form a feature vector of each object region that contains color, texture, and shape, were extracted from each semantic category. Each feature vector consists of 64 color features, 20 texture features, and 24 shape features. A total of 108 features were used to represent an individual object. The PCA method was applied to reduce the feature dimension and to obtain significant features for content analysis. A new feature vector which consisted of 12 new color features, 3 new texture features, and 11 new shape features was obtained. Each feature vector (size = 26) was an input to the MRBFNN. A flow diagram of the training and testing stages is shown in Fig. 1.

The MRBFNN consisted of 8 traditional RBF modules for classifying 8 semantic categories; each traditional RBFNN had 25 hidden neurons. In order to determine an appropriate number of neurons, we tried using various numbers of neurons for the RBFNN. The accuracy of the RBFNNs with various numbers of neurons is shown in

Fig. 7. The figure shows that the RBFNN with 25 neurons achieved the highest accuracy. Thus, each RBFNN used in this study had 25 neurons.

Semantic-based photographic analysis involves finding (1) the semantic category of each region and (2) the image size. Therefore, the percentage of the j -th semantic category can be defined as follows:









$$P_j = \sum_{i=0}^k \frac{W_i^j}{S}, \quad (31)$$

where S denotes the image size, k represents the number of segmented regions, and W_i^j is the size of the i -th segmented region of the j -th semantic category. The accuracy of each testing image is defined as:

$$Accuracy = 1 - \left(\frac{\sum_{i=1}^m A_i}{S} \right), \quad (32)$$

where S denotes the image size, A_i denotes the size of missing area, and m is the number of region that erroneous.

Table 1. Number of categorical images for training and their corresponding color

Semantic Category	Building	Cloud	Sky/Water	Grass	Tree	Flower	Mountain	Sunrise
# of images	39	35	38	30	28	35	35	40
Color								

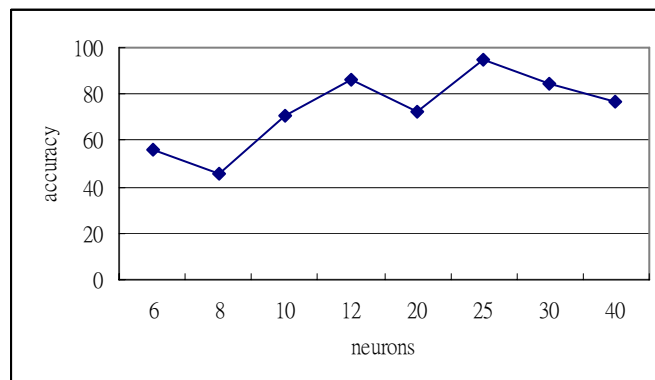


Fig. 7. The accuracy of RBFNN with various numbers of neurons

Fig. 8. and Table 2 show the results of the semantic-based photographic analysis. Fig. 8(f) shows the results of the proposed scheme. The proposed scheme is compared with NonPCA+MRBF, PCA+RBF, and Concurrent SOM (CSOM) [17]. Fig. 8(c) shows the results of NonPCA+MRBF for a training vector with a dimension of 108, Fig. 8(d) shows the results of PCA+RBF for a training vector with a dimension of 26, and Fig. 8(e) shows the results of CSOM for a training vector with a dimension of 26. Table 2 shows the analytic results of each method. The average accuracies of the proposed method, CSOM, NonPCA+MRBF, and PCA+RBF were 94.40%, 75.30%, 72.48%, and 65.67%, respectively. Fig. 8(1) and 8(5) are classified correctly with an accuracy of 100% by the proposed method. Since the content of Fig. 8(5) is simple, all the methods classified the content as sunrise correctly. The image in Fig. 8(7) consists of 4 objects: a building, clouds, grass, and sky/water. The categories of building, cloud, grass and sky/water were classified with 19.31%, 26.33%, 37.52% and 16.82% accuracy, respectively. Although some building regions were misclassified as sky/water or cloud, the classified results are relatively close to human visual perception. The accuracy of the proposed scheme is 85.02%, which is higher than those of the other schemes. Fig. 8(4) is a scenic photo for which a few regions are classified with improper semantic categories since the color and texture features of clouds and sky/water are similar. Fig. 8(6) is an image photographed in the woods. Although some small regions are misclassified as sunrise or flower, the probability of tree is 91.43%. Hence, it is close to human visual perception. Although the results of the image classification have a few errors, the experimental results for the proposed scheme are quite close to human visual perception.

Since we adopt the PCA method to reduce the feature dimension so that the training time of the MRBFNN

can be sped up. Table 3 shows a comparison of the training time of (1) the original feature vectors input in the MRBFNN, and (2) integration of the PCA and MRBFNN. In Fig. 8. and Table 2, the experimental results are better than the others. When the PCA method is used to reduce the feature dimensions, effective and discriminative features can be derived.

Table 2. Classification results of the semantic-based photographic analysis (Unit: %)

Images	Category	Building	Cloud	Sky/Water	Grass	Tree	Flower	Mountain	Sunrise	Accuracy
	Type									
(1)	NonPCA	0.32	0.00	0.00	0.00	54.34	42.62	0.00	2.72	91.29
	RBF	1.78	1.93	0.00	12.81	23.56	44.86	0.00	15.05	57.28
	CSOM	7.94	5.36	37.43	27.29	4.29	0.00	17.69	0.00	49.26
	MRBF	0.00	7.09	0.00	0.00	57.38	35.53	0.00	0.00	100.0
(2)	NonPCA	0.00	0.00	37.73	0.00	1.15	0.00	61.13	0.00	91.96
	RBF	5.88	0.00	0.00	46.31	1.99	31.28	0.00	14.55	45.80
	CSOM	24.48	0.00	40.59	0.00	0.00	0.00	36.35	0.00	76.95
	MRBF	15.95	0.30	0.00	0.00	44.94	33.88	0.00	4.92	82.81
(3)	NonPCA	27.88	0.00	26.05	0.00	1.14	44.93	0.00	0.00	45.74
	RBF	24.79	0.00	0.00	1.55	6.05	24.12	0.00	43.50	46.38
	CSOM	24.48	0.00	40.59	0.00	0.00	0.00	36.35	0.00	78.53
	MRBF	29.52	0.00	0.00	31.31	0.00	24.12	0.00	15.05	99.76
(4)	NonPCA	0.00	0.00	0.00	23.97	58.85	17.18	0.00	0.00	54.31
	RBF	0.00	0.00	0.00	27.86	0.00	37.79	0.00	34.36	67.97
	CSOM	11.08	20.91	42.98	23.96	0.00	0.00	0.00	0.00	82.29
	MRBF	0.11	22.68	0.00	27.86	12.20	37.16	0.00	0.00	99.91
(5)	NonPCA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.0	100.0
	RBF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.0	100.0
	CSOM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.0	100.0
	MRBF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.0	100.0
(6)	NonPCA	5.56	0.00	3.75	0.00	85.96	3.96	0.00	0.78	90.93
	RBF	2.06	0.00	0.00	2.39	93.24	1.28	0.18	0.85	92.50
	CSOM	2.43	4.40	3.15	4.76	82.29	0.00	0.92	0.00	82.29
	MRBF	1.79	2.28	0.00	2.05	91.43	2.06	0.39	0.00	93.30
(7)	NonPCA	21.28	0.00	0.00	0.00	0.00	78.82	0.00	0.00	33.23
	RBF	4.14	0.00	0.00	38.61	18.51	37.49	0.00	1.26	49.78
	CSOM	17.08	21.90	18.84	0.00	34.47	0.00	0.00	0.00	57.81
	MRBF	18.82	31.75	0.00	39.37	0.00	4.09	4.38	1.58	85.02

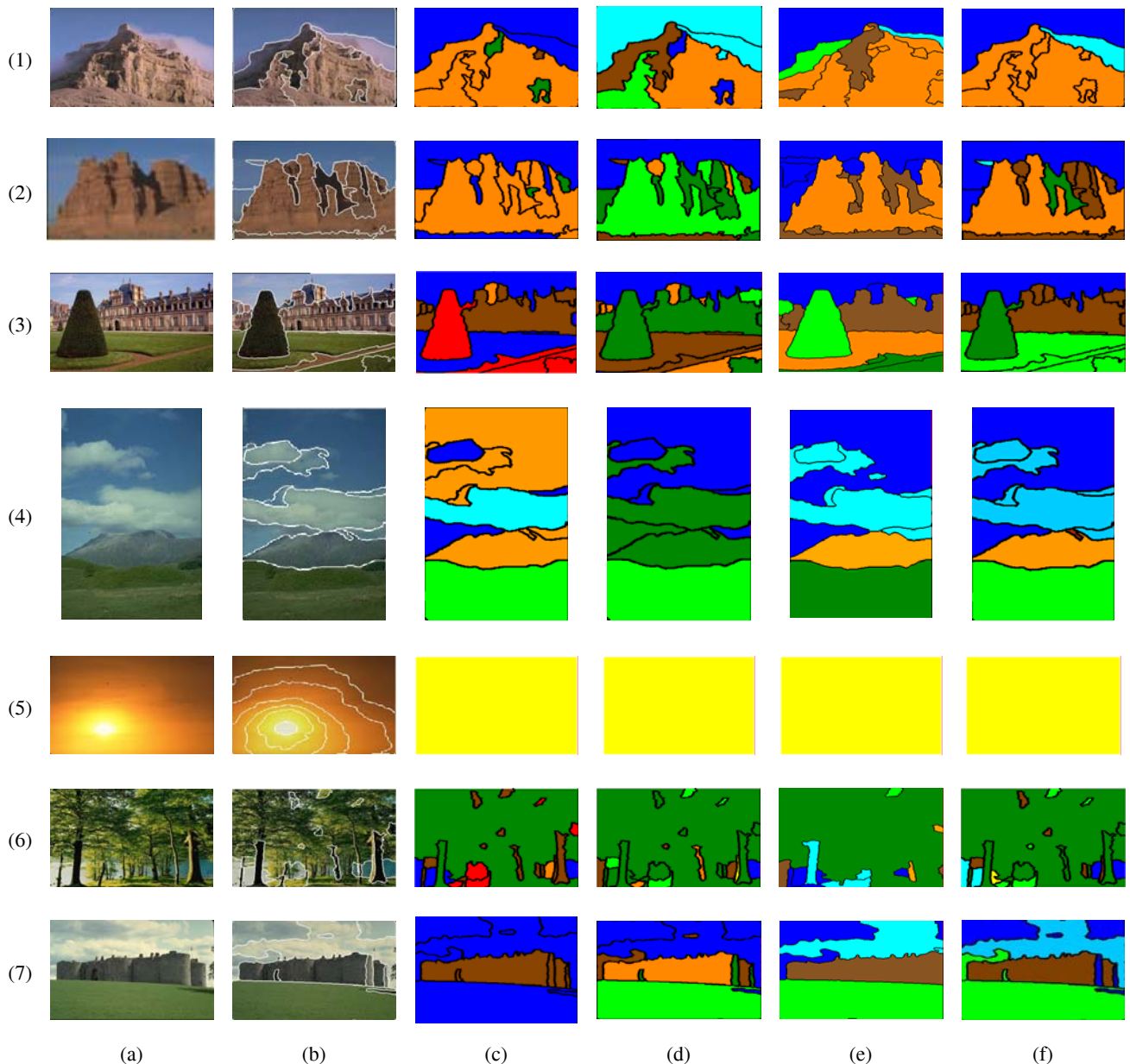


Fig. 8. Classification results of the semantic-based photographic analysis. (a) Original image, (b) Segmented image, (c) Classification result of NonPCA with MRBF, (d) Classification result of PCA with RBF, (e) Classification result of PCA with CSOM, and (f) Classification result of PCA with MRBF

Table 3. Comparison of training time

	Non-PCA with MRBF neural network	PCA with MRBF neural network
Training time	180 minutes	10 minutes and 15seconds

5 Conclusion

A semantic-based content image classification system using low-level image features was proposed to classify various high-level semantic regions. In this paper, we adopted the region-based method rather than the image-based method to derive features. The PCA method was used to obtain more effective and discriminative image features to reduce the feature dimension. An MRBF neural network was proposed for semantic-based image content classification. A combination of SOM and LVQ was used to negate the drawbacks of RBF neural networks in center initialization. 8 traditional RBF neural networks operate independently, and any RBF neural network can be arbitrarily added (or renewed), deleted, or changed. The experimental results of semantic-based

photograph analysis show that the proposed method has a high accuracy rate that is close to human visual perception. Compared to traditional RBF neural networks, the MRBF neural network has a faster learning procedure.

Acknowledgement

This work was supported by the National Science Council of Taiwan, Republic of China, under Grant NSC-96-2221-E-224-070.

References

- [1] C. Faloutsos, R. Barder, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz, "Efficient and Effective Querying by Image Content," *Journal of Intelligent Information Systems*, Vol. 3, pp. 231-262, 1994.
- [2] H.J. Zhang, C.Y. Low, S.W. Smoliar, J.H. Wu, "Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution," *Proceedings of the 3rd ACM International Conference on Multimedia*, pp. 15-24, 1995.
- [3] J.R. Smith and S.F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," *Proceedings of the 4th ACM International Conference on Multimedia*, pp. 87-98, 1997.
- [4] S. Mehrotra, Y. Rui, M.O. Ortega, T.S. Huang, "Supporting Content-Based Image Queries over Images in MARS," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp.632-633, 1997.
- [5] J.Z. Wang, J. Li, G. Wiederhold, "SIMPLicity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, pp. 947-963, 2001.
- [6] Y. Chen, J.Z. Wang, R. Krovetz, "CLUE: Cluster-Based Retrieval of Images by Unsupervised Learning," *IEEE Transactions on Image Processing*, Vol. 14, No. 8, pp. 1187-1201, 2005.
- [7] K. Kuroda and M. Hagiwara, "An Image Retrieval System by Impression Words and Specific Object Names-IRIS," *Neurocomputing*, Vol. 43, No. 1, pp. 259-276, 2002.
- [8] Y. Liu, D.S. Zhang, G.J. Lu, "Region-Based Image Retrieval with High-Level Semantics Using Decision Tree Learning," *Pattern Recognition*, Vol. 41, No. 8, pp. 2554-2570, 2008.
- [9] P.Y. Yin and S.H. Li, "Content-Based Image Retrieval using Association Rule Mining with Soft Relevance Feedback", *Journal of Visual Communication and Image Representation*, Vol. 17, No. 5, pp. 1108-1125, 2006.
- [10] J. Li and J.Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, pp. 1075-1088, 2003.
- [11] J.H. Lim and J.S. Jin, "Combining Intra-Image and Inter-Class Semantics for Consumer Image Retrieval," *Pattern Recognition* Vol. 38, No. 6, pp. 847-864, 2005.
- [12] I. Nwogu and J.J. Corso, "(BP)²: Beyond Pairwise Belief Propagation Labeling by Approximating Kikuchi Free Energies," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [13] Y. Liu, D. Zhang, G. Lu, W.Y. Ma, "Region-Based Image Retrieval with High-Level Semantic Color Names," *Proceedings of IEEE International Conference on Multimedia Modeling Conference*, pp. 180-187, 2005.
- [14] S. Chitroub, A. Houacine, B. Sansal, "Principal Component Analysis of Multispectral Images using Neural Network," *Proceedings of 1st ACS/IEEE International Conference on Computer Systems and Applications*, pp. 89-95, 2001.
- [15] W.Y. Ma and B. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases," *Proceedings of IEEE International Conference on Image Processing*, pp. 568-571, 1997.

- [16] C. Carson, S. Belongie, H. Greenspan, J. Malik, "Blobworld: Image Segmentation using Expectation-Maximization and Its Application to Image Querying," *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 24, No. 8, pp. 1026-1038, 2002.
- [17] T.W.S. Chow and M.K.M. Rahman, "A New Image Classification Technique Using Tree-Structured Regional Features," *Neurocomputing*, Vol. 70, No. 4-6, pp. 1040-1050, 2007.
- [18] M.K. Muezzinoglu and J.M. Zurada, "RBF-Based Neurodynamic Nearest Neighbor Classification in Real Pattern Space," *Pattern Recognition*, Vol. 39, No. 5, pp. 747-760, 2006.
- [19] S.B. Park, J.W. Lee, S.K. Kim, "Content-Based Image Classification Using Neural Network," *Pattern Recognition*, Vol. 25, No. 3, pp. 287-300, 2004.
- [20] C.F. Tsai, K. McGarry, J. Tait, "Image Classification Using Hybrid Neural Network," *Proceedings of 26th Annual International ACM SIGIR Conference Research and Development in Information Retrieval*, pp. 431-432, 2003.
- [21] D. Wang, J.S. Lim, M.M. Han, B.W. Lee, "Learning Similarity for Semantic Images Classification," *Neurocomputing*, Vol. 67, pp. 363-368, 2005.
- [22] F. Ros, M. Pintore, A. Deman, J.R. Chrétien, "Automatic Initialization of RBF Neural Networks," *Chemometrics and Intelligent Laboratory Systems*, Vol. 87, No. 1, pp. 26-32, 2007.
- [23] A.C. Rencher, *Multivariate Statistical Inference and Applications*, John Wiley, New York, 1998.
- [24] N. Vaswani and R. Chellappa, "Principal Components Null Space Analysis for Image and Video Classification," *IEEE Transactions on Image Processing*, Vol. 15, No. 7, pp. 1816-1830, 2006.
- [25] Y. Deng and B.S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 8, pp. 800-810, 2001.
- [26] R. Haralick, K. Shanmugam, I. Dinstein, "Texture Feature for Image Classification," *IEEE Transactions on System, Man, and Cybernetics*, Vol. 8, No. 6, pp. 610-621, 1973.
- [27] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 837-842, 1996.
- [28] Y.C. Cheng and S.Y. Chen, "Image Classification Using Color, Texture and Regions," *Image and Vision Computing*, Vol. 21, No. 9, pp. 759-776, 2003.
- [29] F. Mahmoudi, J. Shanbehzadeh, A.M. Eftekhari-Moghadam, H. Soltanian-Zadeh, "Image Retrieval based on Shape Similarity by Edge Orientation Autocorreslogram," *Pattern Recognition*, Vol.36, No. 8, pp. 1725-1736, 2003.
- [30] O. Chapelle, P. Haffner, V.N. Vapnik, "Support Vector Machines for Histogram-Based Image Classification," *IEEE Transactions on Neural Network*, Vol. 10, No. 5, pp. 1055-1064, 1999.
- [31] X. Ma and D. Wang, "Semantics Modeling Based Image Retrieval System Using Neural Networks," *Proceedings of IEEE International Conference on Image Processing*, pp. 1165-1168, 2005.
- [32] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, H.J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Transactions on Image Processing*, Vol. 10, No. 1, pp. 117-130, 2001.