Feature Encoding for Unsupervised Segmentation of Color Images

N. Li and Y. F. Li, *Senior Member, IEEE*

***Abstract*—In this paper, an unsupervised segmentation method using clustering is presented for color images. We propose to use a neural network based approach to automatic feature selection to achieve adaptive segmentation of color images. With a self-organizing feature map (SOFM), multiple color features can be analyzed, and the useful feature sequence (feature vector) can then be determined. The encoded feature vector is used in the final segmentation using fuzzy clustering. The proposed method has been applied in segmenting different types of color images, and the experimental results show that it outperforms the classical clustering method. Our study shows that the feature encoding approach offers great promise in automating and optimizing the segmentation of color images.**

***Index Terms*—Automatic feature selection, color spaces, clustering, unsupervised segmentation.**

I. Introduction

SINCE color provides important cues for image processing and object recognition, color images are used in many applications. However, the processing of color images presents special challenges since they are multidimensional. This is particularly the case for segmentation of color images. Conventional image segmentation relies mainly on a technique to detect the uniformity of the feature values of the image pixels [1], [2]. For color image segmentation, different approaches have been explored. Among them, neural network based approach has received considerable attention [3]–[5], [9]. Using a neural network, Littmann designed a local linear mappings (LLM) network based on LLMs for segmentation [3]. The network architecture is closely related to the self-organizing maps. Campadelli [4] proposed two different methods based on Huang's idea of using Hopfield networks [5]. With their network, spatial information can be taken into account. However, this method requires that the number of clusters be correctly estimated by histogram analysis to determine the network structure and its initialization.

Clustering algorithms based on unsupervised classification have been widely used for color image segmentation [6], [7], [15]–[18]. Kehtarnavaz [6] employed multiscale clustering for grouping similar color pixels in a color image. Panjwani [7] developed a segmentation algorithm based on hierarchical clustering for segmenting texture color images. Cramariuc [18] combined the clustering in the color space with region growing in the image space for achieving automatic segmentation. Without the *priori* information of the image, the clustering methods only require the number of clusters and then self-organize to generate the clusters. If the image contains only homogenous regions, the clustering methods in the color space are sufficient to handle the problem. Unfortunately, most clustering algorithms suffer from the enormous costs of computation if the data of the class are scattered over the whole color space. The high computational cost has thus limited the practical applications of this segmentation approach. A well-suited color space needs to be chosen where color and intensity are coded independently.

Since the effectiveness of color image segmentation depends on the color reference system used, most algorithms designed are only suitable for use in a specific color space such as RGB [3], [4], “IJK” or KL_RGB [4], [9], CIEL*a*b* [15], or HIS [17]. Using K–L transformation of RGB color feature, Ohta was able to segment eight kinds of color images by recursive thresholding [8]. A major disadvantage of RGB system is the dependency of the three features on the light intensity [4], [19]. This weakness can be overcome in other representation system where color and intensity are represented independently. Kehtarnavaz [6] chose the p–q geodesic space for clustering since the geodesic space provides perceived color shifts throughout the space. Littmann [3] used the YUV system because of its continuous representation of the color components. However, the experimental results showed that using RGB data provided by the cameras can give the best results. To achieve lower classification error and higher processing speed, Verikas [9] chose the “IJK” color space when using modular neural network for segmentation. The features, I, J, and K, are given by the eigensolution of the covariance matrix of the variables R, G, and B that is similar to those derived by Ohta [8]. Gauch [10] studied three different algorithms for segmentation in four different color spaces, RGB, YUV, HLS, and CIE L∗a∗b∗ spaces. The experimental results were not totally consistent. As a result, they could hardly draw a conclusion on the influence of the chosen color spaces. The MCIS algorithm developed by Liu [11] does not need *a priori* information and is noise resistant. However, when comparing the influences of different color spaces on the segmentation results, they found that no single color space chosen could produce good segmentation results for all the testing color images in using the MCIS algorithm. Similar observations were also made by other researchers [3], [4], [9]–[12].

Manuscript received April 16, 2000; revised November 5, 2001. This work was supported by a grant from the Research Grants Council of Hong Kong under Project CityU1136/98E (9040368). This paper was recommended by Associate Editor V. Murino.

N. Li is with the Department of Electronic Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

Y. F. Li is with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong.

Digital Object Identifier 10.1109/TSMCB.2003.811120

Therefore, it is important to extract well-suited color features from different color systems for color image processing [13], [14] and segmentation. One way to achieve this is to execute the segmentation using different sets of color features and then to compare the results to determine the effective sets of color features [3], [8], [11]. This allows us to predefine the sets of color features. This is apparently a time consuming process. A desirable way is to determine a set of well-suited color features first and then segment the color images using this set of color features [6], [9], [15], [19]. Our proposed approach here belongs to the latter. In our method, the determination of the effective color features depends on the analysis of various color features from each testing color image via the designed feature encoding. This is different from the previous methods like Verikas method [9], where they chose a set of features for all testing color images since the feature set was suitable for their proposed network. The difference here is that our method is adaptable for segmenting different types of color images. With our method, the segmentation is based on the contributions of color features rather than the choice of a particular color space [6], [9], [15]. Here, a self-organizing feature map (SOFM) is used in the feature encoding so that it can self-organize the effective features for different color images. Fuzzy clustering is applied for the final segmentation when the well-suited color features and the initial parameters are available. With efficient feature analysis, our proposed method improves on the performance of clustering for color image segmentation.

This paper is organized as follows: In Section II, the feature encoding by SOFM network in determining the well-suited feature vectors for segmentation is proposed. Section III presents the results of color image segmentation using our method. Comparisons of our results with those using classical clustering are also given. To verify the effectiveness of our proposed method, Section IV gives an evaluation of the performance of the proposed algorithm. The conclusions are given in Section V.

II. Feature Encoding

It is well known that all colors are perceived as combinations of three primary colors: red, green, and blue (RGB). Almost all of today's color image acquisition devices output images in RGB space. However, RGB space is not always ideal for different segmentation algorithms as will be shown in the experiments. A good segmentation algorithm requires the selection of a proper color space. We propose here to select the features for segmentation without being limited by any particular color space. We will show that suitable color feature vectors can be automatically extracted for segmentation via the feature encoding.

The feature encoding here is accomplished here via a neural network based approach. The feature encoder network consists of a number of feature vector units in the input layer and a feature map unit in the output layer, as shown in Fig. 1. Each neuron in the input layer represents a feature and has its weights associated with the connections from itself to each neuron in the output layer. Assume that p features are to be encoded. Then, the number of neurons in the input layer is p. Feature vectors are extracted from the input image. The feature map units are represented in a two-dimensional (2-D) output space where the neurons are arranged as a 2-D array. The encoded feature vectors Q can then be determined from the resulting SOFM. For color images, features can be extracted from different color spaces such as RGB, SCT, YUV, HLS, L*a*b* spaces etc. To compress the input image data, a receptive field can be adopted which is a movable, nonoverlapping window. For an $m \times n$ color image with p-dimensional feature vectors, if the window size is $k \times l$, then the number of input sample data N is compressed by $k \times l$ times. For a set of patterns represented by these compressed feature vectors, assuming each pattern $x_i = \{f1, f2, f3, \ldots, fp\}^T$, then the training sample data, i.e., the set of patterns, is represented as $X = \{x_1, x_2, x_3, \ldots, x_N\}^T$.

Here, we use the SOFM to train the input feature vectors in the encoder network. A mapping from the feature space to the spatial space is accomplished via the SOFM through its spatial self-organizing capability. During a training cycle, only one of the output neurons, i.e., the winner, can fire at a time. That is, for each input vector, the winning output neuron will assign the input vector an appropriate class. The winner is determined by the minimal distance criterion. Since the network is self-organizing, the synaptic weight vector $\boldsymbol{w}_j$ needs to change with the input vector $\boldsymbol{x}$. We modify the Hebbian hypothesis by including a nonlinear forgetting term, $g(y_j)\boldsymbol{w}_j$, where $\boldsymbol{w}_j$ is the synaptic weight vector of neuron j, and $g(y_j)$ is some positive scalar function of its response y_j. The only requirement imposed on the function $g(y_j)$ is that the constant term in the series expansion of $g(y_j)$ is zero. Then, we have

$$g(y_j) = 0, \qquad \text{for } y_j = 0, \quad \text{and for all } j. \tag{1}$$

Using such a function, we may then define the computational map of SOFM algorithm given by the following differential equation

$$\frac{d\vec{w}_j}{dt} = \eta y_j \vec{x} - g(y_j)\vec{w}_j, \qquad j = 1, 2, \ldots, N \tag{2}$$

where t denotes the continuous time, and η is the learning rate of the algorithm.

In practice, (2) can be simplified further. If the input vector $\boldsymbol{x}$ changes at a rate that is low compared to that of the synaptic weight vector $\boldsymbol{w}_j$ for all j, we may justifiably assume that due to the clustering effect, the response y_j of neuron j is at either a low or high saturation value depending on whether neuron j is active or not. In addition, when neuron j is in the temporarily activated state, the function takes the middle value. Thus, by identifying the activity of the neighborhood function $\Lambda_{i(x)}$, we have

$$y_j = \begin{cases} 1, & \text{neuron } j \text{ is active} \\ 0.5, & \text{neuron } j \text{ in the temporary case} \\ 0, & \text{neuron } j \text{ is inactive.} \end{cases} \tag{3}$$

Similarly, we may express the function $g(y_j)$ as

$$g(y_j) = \begin{cases} \alpha, & \text{neuron } j \text{ is active} \\ 0.5\alpha, & \text{neuron } j \text{ in the temporary case} \\ 0, & \text{neuron } j \text{ is inactive} \end{cases} \tag{4}$$

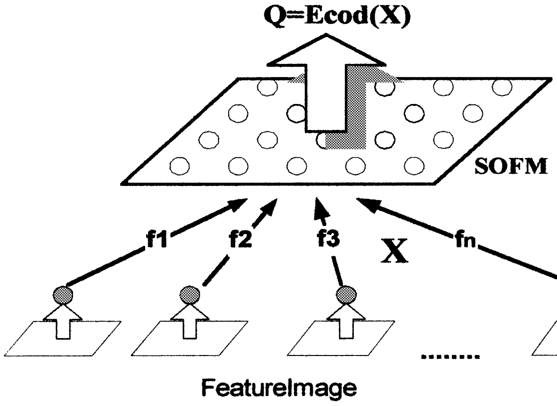


Fig. 1. Feature encoding network.

where α is a positive constant. Accordingly, we may simplify (2) as

$$\frac{d\vec{w}_j}{dt} = \eta\vec{x} - \alpha\vec{w}_j, \quad \text{neuron } j \text{ is inside the neighborhood.} \tag{5}$$

Without loss of generality, we may use the same scaling factor for the input vectors $\boldsymbol{x}$ and the weight vectors $\boldsymbol{w}_j$. In other words, putting $\eta = \alpha$, we can simplify equation (5) further as

$$\frac{d\vec{w}_j}{dt} = \eta(\vec{x} - \vec{w}_j), \quad \text{neuron } j \text{ is inside the neighborhood.} \tag{6}$$

According to (6), we know that the weight vectors $\boldsymbol{w}_j$ tend to follow the input vectors $\boldsymbol{x}$ as time t increases. Given the synaptic weight vectors $\boldsymbol{w}_j(\boldsymbol{t})$ of neuron j at discrete time t, we may compute the updated value $\boldsymbol{w}_j(\boldsymbol{t}+\boldsymbol{1})$ at time $t+1$ by

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \eta(t)\Lambda_{i(x)}(t)\left(\vec{x} - \vec{w}_j(t)\right), \quad j \in \Lambda_{i(x)}(t). \tag{7}$$

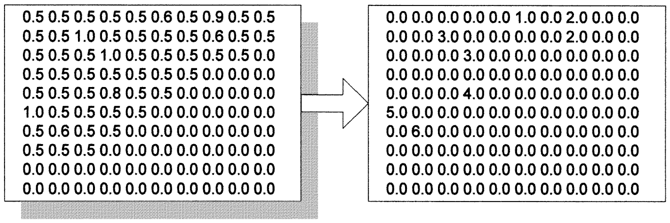
Here, $\Lambda_{i(x)}(t)$ is the neighborhood function around the winning neuron i at time t, and $\eta(t)$ is the corresponding value of the learning rate.

The network is then trained by the self-organizing competitive learning algorithm. The training process continues until the encoder network converges. A cost function $E(t)$, referred to as Lyapunov E function, is defined to determine the convergence of the encoder network:

$$E(t) = -\tfrac{1}{2}\sum_{ij}\sum_{\mu=1}^{n}\Lambda(i, i^*, \mu)\left(x_i^\mu - w_{ij}(t)\right)^2 \tag{8}$$

where $\mu_i(n)$ is the update counter for neuron i, and x_i^μstands for the input vectors. After minimizing the cost function, the network trained is in a stable state. The weights associated with the connections from each feature to the output neuron will also become fixed values.

Neurons in the feature map are arranged in a regular geometric structure before training. When training the encoder network with different feature vectors, each time at the start, one type of neural cells in the output layer is excited by the input samples, with the corresponding positions of the cells in the feature map changed. In the end, all the neurons are rearranged in the feature map in such a way that the neurons representing the same characteristics of the input feature vectors are grouped, whereas the neurons representing different characteristics of the input feature vectors are separated from each other. These statistical characteristics in the feature map can be also determined by the output responses of the neurons.



0.5	0.5	0.5	0.5	0.5	0.6	0.5	0.9	0.5	0.5
0.5	0.5	1.0	0.5	0.5	0.5	0.5	0.6	0.5	0.5
0.5	0.5	0.5	1.0	0.5	0.5	0.5	0.5	0.5	0.0
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.0	0.0	0.0
0.5	0.5	0.5	0.8	0.5	0.5	0.0	0.0	0.0	0.0
1.0	0.5	0.5	0.5	0.5	0.0	0.0	0.0	0.0	0.0
0.5	0.6	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0
0.5	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

0.0	0.0	0.0	0.0	0.0	1.0	0.0	2.0	0.0	0.0
0.0	0.0	3.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0
0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 2. Feature map for an image "SQUARE3."

During the feature encoding, the output layer maps the final response of the neural net to give the same output $\boldsymbol{y}$ if any of the nodes from the same cluster fires. Similar input vectors always cause nodes from a local neighborhood to fire in response to the patterns from a class. We use the output response y_j of neuron j to depict the topology of a cluster in the feature map. Equations (2) and (3) show that if a neuron is active, it outputs 1; otherwise, it outputs 0. If a neuron is in the temporarily activated state, it outputs 0.5. After the network has converged at time n, we calculate the mean of all the output responses z_j of neuron j

$$z_j = \frac{1}{n}\sum_{j=1}^{n} y_j = \begin{cases} 0, & \text{neuron } j \text{ is always no active} \\ 0.1\text{–}0.4, & \text{neuron } j \text{ is in no activity} \\ & \quad \text{or temporary} \\ 0.5, & \text{neuron } j \text{ is always temporary} \\ 0.6\text{–}0.9, & \text{neuron } j \text{ is in temporary} \\ & \quad \text{or activity} \\ 1, & \text{neuron } j \text{ is always active.} \end{cases} \tag{9}$$

Furthermore, we simplify the topology of the clusters in the feature map as a region map separated by the boundaries of the neuron groups. This results in several regions in the SOFM. Fig. 2 shows the region maps of a color image after the network has converged. The distribution of region map on the left is the mean of all the output responses, z_j. The region map on the right characterizes the topology of six clusters in the feature map.

The effect of updating equation (7) is the motion of the synaptic weight vectors $\boldsymbol{w}_i$ of the winning neuron i toward the input vectors $\boldsymbol{x}$. Upon repeated training, the synaptic weight vectors tend to follow the distributions of the input vectors due to its neighborhood updating. Therefore, the algorithm leads to a topological ordering of the feature map in the output space. On the feature map, the distributions of the weights associated with each connection from the input feature vectors to the output neurons reflect the contributions of the input feature vectors to the clustering. For $p \times q$ neurons in a 2-D output layer of the encoder network, all the weights W_{ij}, associated with a connection from the jth feature to ith neuron, are obtained. An encoded feature parameter A_f^c is defined to represent the

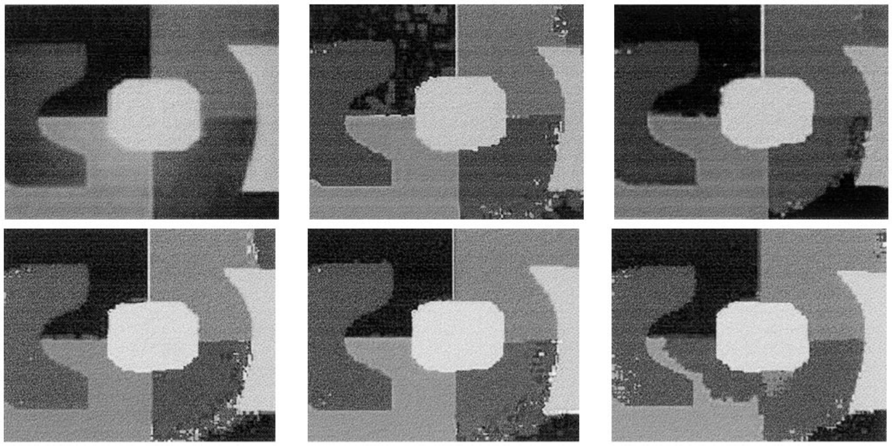


Fig. 3. Segmentation results based on feature encoding. (a) Original image "SQUARE3" with RGB description. (b) Segmentation using feature vectors $\{f1,\, f4,\, f15\},\ \delta = 0.9$. (c) Segmentation using feature vectors $\{f1,\, f4,\, f12,\, f15\},\ \delta = 0.6$. (d) Segmentation using feature vectors $\{f1,\, f2,\, f4,\, f12,\, f13,\, f15\},\ \delta = 0.5$. (e) Segmentation using feature vectors $*\{f1,\, f2,\, f4,\, f11,\, f12,\, f13,\, f15\},\ \delta = 0.3$. (f) Segmentation using feature vectors $\{f1,\, f2,\, f3,\, f4,\, f7,\, f11,\, f12,\, f13,\, f14,\, f15\},\ \delta = 0.2$.

contribution of feature f to the clustering from feature f to all the neurons in region c in the feature map. This is given as follows:

$$A_f^c = \frac{1}{r_c} \sum_{i=1}^{r_c} w_{if}, \qquad f = 1,\, 2,\, \ldots,\, p;\ c = 1,\, 2, \ldots,\, k \tag{10}$$

where r_c represents r neurons in region c, and w_{if} is the weight associated with the connection from feature f to neuron i. The contribution of each feature to the clustering is evaluated by the variation D_f, which is derived from the encoded feature parameter A_f^c as follows:

$$d_f = \frac{1}{k} \sum_{c=1}^{k} \left[A_f^c - \overline{A_f} \right]^2, \qquad f = 1,\, 2,\, \ldots p \tag{11}$$

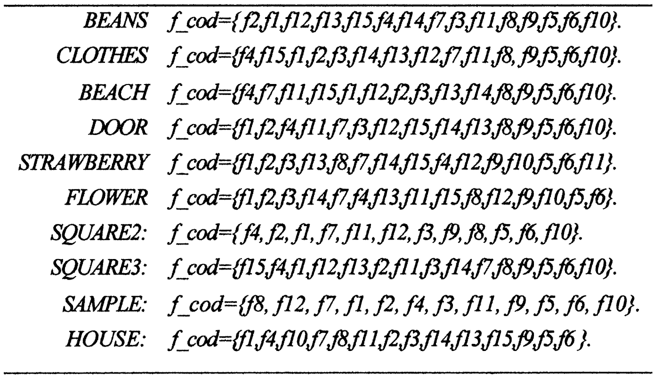
where

$$\overline{A_f} = \frac{1}{k} \sum_{c=1}^{k} A_f^c, \qquad f = 1,\, 2,\, \ldots p. \tag{12}$$

If a feature has a strong influence on the clustering, its variation will be larger than others. A definite encoded feature sequence is then obtained. The above approach for feature encoding is referred to as the Encoder Segmented Neural Network (ESNN). In practice, a normalized coefficient of the variation is defined as follows:

$$D\_coef = \frac{D_f}{\max\{D_f | f = 1,\, 2,\, 3,\, \ldots,\, p\}}. \tag{13}$$

A feature, which is the constituent of a feature vector, is ultimately determined by its value of D_coef. The segmentation here is achieved using fuzzy clustering, but it is an improved version of the classical Fuzzy C-Means (FCM) algorithm. Using our method, the final color image segmentation is accomplished by the encoded feature-based FCM, referred to as EFFCM algorithm [20]. The average color for each sub-region is calculated based on the original image after applying our method. RGB representation of color is then assigned to each pixel of the subregion since we need to display the segmented color image. Once the pixel of each region is labeled with the new RGB value, the segmented image can be displayed.



BEANS	f_cod={ f2,f1,f12,f13,f15,f4,f14,f7,f3,f11,f8,f9,f5,f6,f10}.
CLOTHES	f_cod={f4,f15,f1,f2,f3,f14,f13,f12,f7,f11,f8, f9,f5,f6,f10}.
BEACH	f_cod={f4,f7,f11,f15,f1,f12,f2,f3,f13,f14,f8,f9,f5,f6,f10}.
DOOR	f_cod={f1,f2,f4,f11,f7,f3,f12,f15,f14,f13,f8,f9,f5,f6,f10}.
STRAWBERRY	f_cod={f1,f2,f3,f13,f8,f7,f14,f15,f4,f12,f9,f10,f5,f6,f11}.
FLOWER	f_cod={f1,f2,f3,f14,f7,f4,f13,f11,f15,f8,f12,f9,f10,f5,f6}.
SQUARE2:	f_cod={ f4, f2, f1, f7, f11, f12, f3, f9, f8, f5, f6, f10}.
SQUARE3:	f_cod={f15,f4,f1,f12,f13,f2,f11,f3,f14,f7,f8,f9,f5,f6,f10}.
SAMPLE:	f_cod={f8, f12, f7, f1, f2, f4, f3, f11, f9, f5, f6, f10}.
HOUSE:	f_cod={f1,f4,f10,f7,f8,f11,f2,f3,f14,f13,f15,f9,f5,f6 }.

Fig. 4. Encoded feature sequence for each color image.

III. Experimental Results

To verify the effectiveness of our proposed method, we used some color images for testing in the experiments. The experiments were conducted using C++ and MATLAB programming on a Pentium III 450 platform with 128 MB memory. Each image is originally described in RGB space. Fifteen features are adopted from the intensities in RGB, SCT, YUV, HIS, and CIE L∗a∗b∗ color spaces for the color images. In our encoding network, the input layer consists of 15 neurons. The output layer consists of 100 neurons arranged as a 2-D space map with 10 × 10 neuron structure. The input, a set of sample data for feature encoding, is constructed as an $n \times 15$-integer value matrix whose row vector consists of 15 features associated with a pixel. The weights associated with the connections from each neuron between the input layer and the output layer is constructed as a 100 × 15 matrix. In the experiments, we set the neighborhood N_c as 3. The feature encoding is considered as converged when the error coefficient is below to 0.0001. Some experimental results are given in Figs. 3–14.

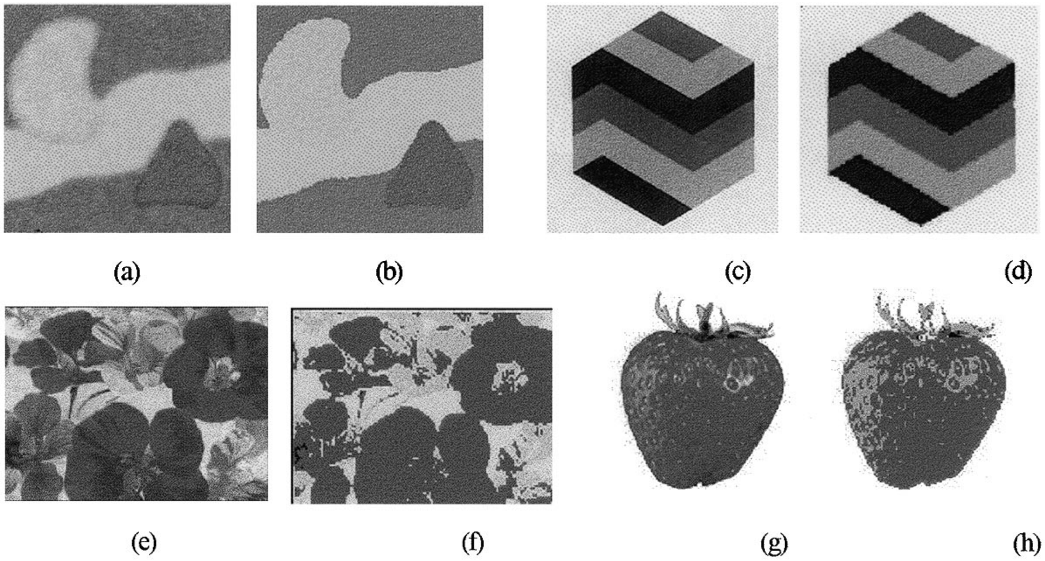


Fig. 5. Segmentation results using ESNN-based method. (a) Original image "SAMPLE" with RGB description. (b) Segmentation with four classes using encoded features $\{f7,\ f8,\ f12\}, \delta = 0.35$. (c) Original image "SQUARE2" with RGB description. (d) Segmentation with four classes using encoded features $\{f2,\ f4\}$, $\delta = 0.6$. (e) Original image "FLOWER" with RGB description. (f) Segmentation with three classes using encoded features $\{f1,\ f2,\ f3\}, \delta = 0.6$. (g) Original image "STRAWBERRY" with RGB description. (g) Segmentation with three classes using encoded features $\{f1,\ f2\}, \delta = 0.6$.

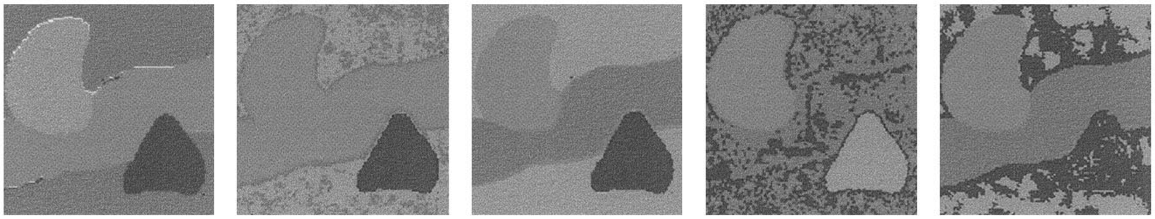


Fig. 6. Segmentation results of image "SAMPLE" using FCM method in different color spaces. (a) RGB space. (b) SCT space. (c) YUV space. (d) HLS space. (e) L$*$a$*$b$*$ space.

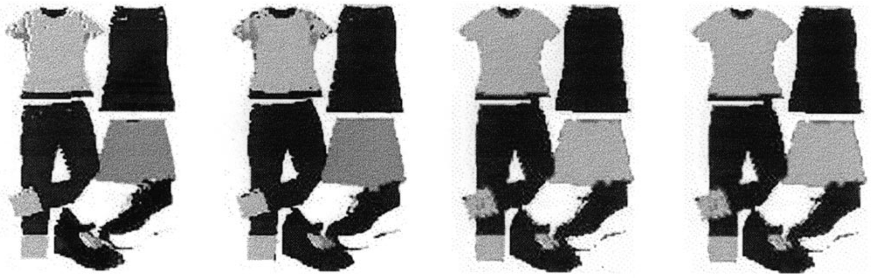


Fig. 7. Segmentation results of image "CLOTHES" using FCM method in different color spaces. (a) RGB space. (b) SCT space. (c) YUV space. (d) HLS space.

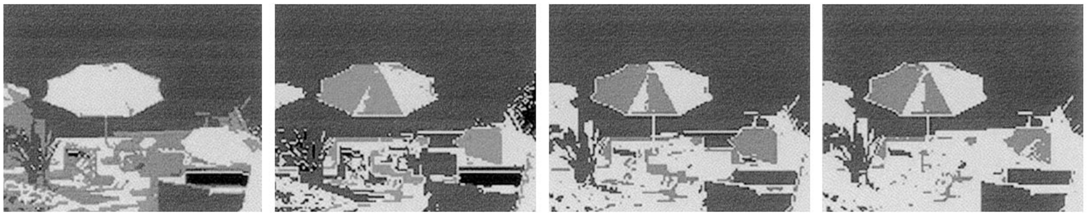


Fig. 8. Segmentation results of image "BEACH" using FCM method in different color spaces. (a) RGB space. (b) SCT space. (c) YUV space. (d) HLS space.

The ESNN-based clustering algorithm first determines the initial number of clusters and the feature vectors for the fine segmentation. Here, the initial number of clusters c_0 is determined online. Fig. 2 shows the result of the feature map for a color image "SQUARE3" when the feature encoder network has converged. The "0" on the left half of the figure indicates a nonactivated neuron, whereas "1.0" indicates an activated neuron. An output value ranging from 0.6 to 0.9 indicates that the neuron

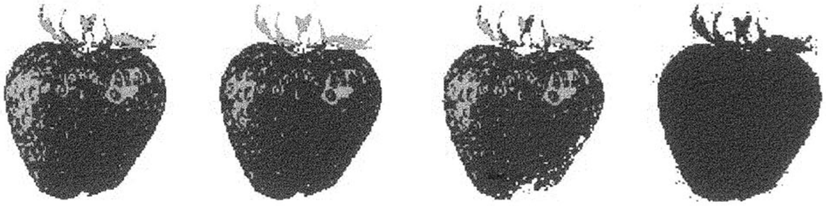


Fig. 9. Segmentation results of image “STRAWBERRY” using FCM method in different color spaces. (a) RGB space. (b) YUV space. (c) HLS space. (d) L$*$a$*$b$*$ space.

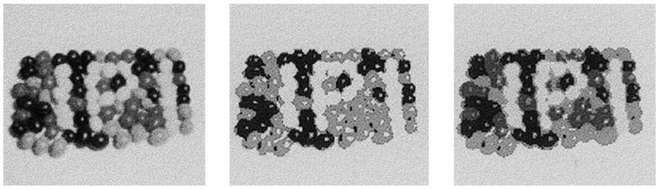


Fig. 10. Comparison of the results using FCM and ESNN method. (a) Original image “BEANS” with RGB description. (b) Segmentation using FCM method in RGB space. (c) Segmentation using ESNN method using feature vector $\{f1,\ f2\}$, $\delta = 0.6$, *five classes*.

is between the states of temporary and activated, in which case, the neuron has certain probability to be clustered into the group with its neighbors. A threshold value for each region boundary should be 0.5, representing that the neuron is in the temporary state, subject to a tolerance of ± 0.1. The corresponding region distributions on the SOFM are then determined as shown on the right half of Fig. 2. Here, c_0, the number of regions, is 6 for the color image “SQUARE3.”

The determination of the feature vectors depends on the contributions of features to the clustering. Corresponding to the feature map of image “SQUARE3,” the encoded feature parameters A_f^c, the variations of each feature D_f, and the normalized coefficients D_coef are then calculated, as shown in Table I. Note that $f1$–$f3$ are the features from RGB space; $f4$–$f6$ are the features from SCT space; $f7$–$f9$ are the features from YIQ space; $f10$–$f12$ are the features from HIS space; and $f13$–$f15$ are the features from CIEL*a*b* space. For clarity of explanation, the encoded definite feature sequence can be given by the values of the normalized coefficients of feature variations, i.e.,

$$f\_cod = \{f15,\ f4,\ f1,\ f12,\ f13,\ f2,\ f11,\ f3,\ f14 \\ f7,\ f8,\ f9,\ f5,\ f6,\ f10\}$$

for image “*SQUARE3.* With different threshold values for the normalized coefficients of feature variations, different feature vectors can be derived. The initial vectors of the center of clusters are set to random values. We then determine a positive definite matrix $\boldsymbol{A}$, reflecting the relationship between the feature and the center of the clusters, as an identity matrix. A value $T = 0.001$ is used as the threshold for judging the convergence of the EFFCM algorithm. Depending on the defuzzification method used, each pixel is uniquely assigned to the class that has the maximum membership function.

The segmentation results using our ESNN method for image “SQUARE3” is shown in Fig. 3, where several online results are given when different feature sequences encoded are used for the segmentation. The feature vectors, i.e., $\{f1,\ f4,\ f15\}$, $\{f1,\ f4,\ f12,\ f15\}$, $\{f1,\ f2,\ f4,\ f12,\ f13,\ f15\}$, $\{f1,\ f2, f4,\ \ f11,\ f12,\ \ f13, f15\}$, and $\{f1,\ f2,\ f3, f4,\ f7,\ f11, f12,\ f13,\ f14,\ f15\}$ were derived when the normalized coefficient δ was 0.9, 0.6, 0.5, 0.3, and 0.2. The result marked with stars $*$ in Fig. 3 is the final result for the segmentation when using the encoded feature vector $\{f1,\ f2,\ f4,\ f11,\ f12,\ f14,\ f15\}$. The actual (final) number of clusters c for image “SQUARE3” is seven.

Experiments with other color images are then conducted using our method via feature encoding. The calculation of D_coef for each color image is presented in Table II. The encoded definite feature sequences are given in Fig. 4. It can be seen that for different color images, the encoded feature sequences are different. In addition, with different thresholds for the normalized coefficients of feature variations, different feature vectors were derived for the segmentation.

Fig. 5 shows some final results of our method for the segmentation. In the final segmentation for image “SAMPLE,” the encoded feature vector $\{f7,\ f8,\ f12\}$ was obtained with $\delta = 0.35$, while for image “SQUARE2” a feature vector $\{f2,\ f4\}$ was encoded with $\delta = 0.6$. It can be inferred that the optimal feature vector suitable for the segmentation varies with different types of images. It should be noted that in our ESNN-based method, no preference is given to any particular color space. Rather, the encoder network can determine the best combinations of features (or spaces) for the application in concern.

We also conducted experiments in comparing our method with a traditional clustering method (the FCM algorithm) in different color spaces. With FCM method, the pixels belonging to a valid class are clustered. A cluster is determined if the maximum value of the membership function is below the threshold T. Figs. 6–9 shows the results of segmentation using FCM method in different color spaces. In Fig. 6, the results in segmenting image “SAMPLE” are unsatisfactory in SCT space, HLS space, and L$*$a$*$b$*$ space. In Fig. 7, the results with image “CLOTHES” in SCT space, HLS space, and YUV space show some mixing up of the trousers, shoes, and long skirt in the segmented images. Furthermore, the results in YUV space and HLS space does not provide distinguishable segmentation of the short skirt and T-shirt. In these two examples, the results in RGB space are relatively better than those in other spaces. In Fig. 8, none of the segmentation results for image “BEACH” in the five color spaces turns out to be satisfactory. In Fig. 9, the result with image “STRAWBERRY” in L$*$a$*$b$*$ space is not satisfactory, while that in YUV space is better. The results show that using FCM method with a single color space, the segmentation results may not always be the best or

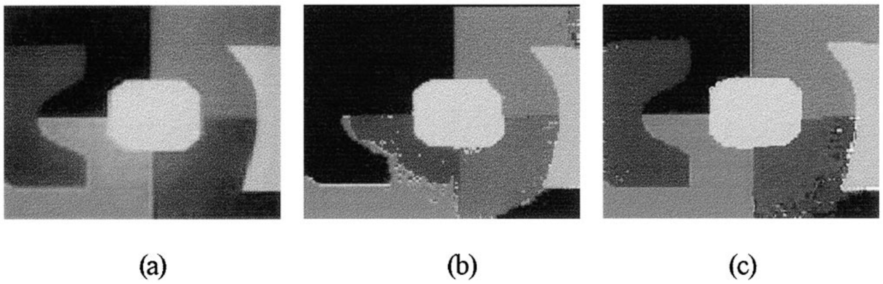


Fig. 11. Comparison of the results using FCM and ESNN method. (a) Original image "SQUARE3" with RGB description. (b) Segmentation using FCM method in RGB space. (c) Segmentation using ESNN method using feature vector $\{f1,\ f2,\ f4,\ f11,\ f13,\ f15\}, \delta = 0.3$, *seven classes*.

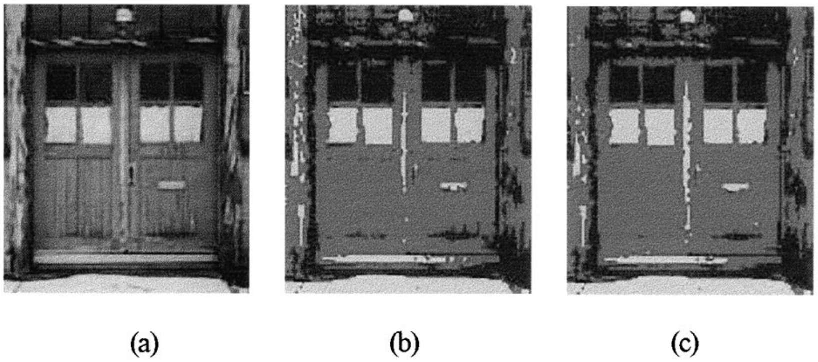


Fig. 12. Comparison of the results using FCM and ESNN method. (a) Original image "DOOR" with RGB description. (b) Segmentation using FCM method in RGB space. (c) Segmentation using ESNN method using feature vectors $\{f1,\ f2\}, \delta = 0.3$, *three classes*.

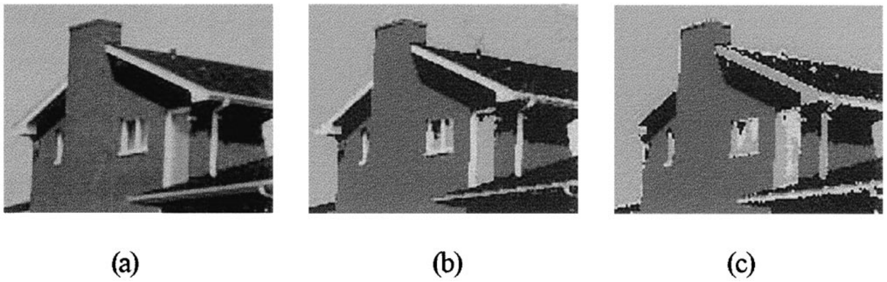


Fig. 13. Comparison of the results using FCM and ESNN method. (a) Original image "HOUSE" with RGB description. (b) Segmentation using on FCM method in RGB space. (c) Segmentation using ESNN method with feature vectors $\{f1,\ f4,\ f7,\ f8,\ f10\}, \delta = 0.3$, *four classes*.

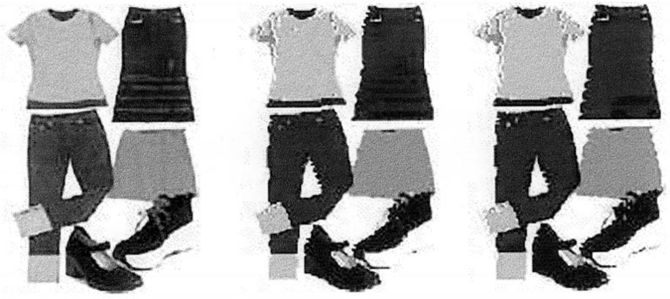


Fig. 14. Comparison of the results using FCM and ESNN method. (a) Original image "CLOTHES" with RGB description. (b) Segmentation using FCM method in RGB space. (c) Segmentation using ESNN method with feature vectors $\{f1,\ f2,\ f3,\ f4,\ f15\}$, *five classes*.

even acceptable for different types of images. Furthermore, this method requires prior knowledge of some parameters for the segmentation including the number of clusters c, the membership weighting exponent m, the threshold T for the membership function, etc. Therefore, the performance of the traditional clustering method relies heavily on the knowledge available and the luck in choosing a correct color space for the segmentation.

Using a traditional clustering method, it is possible to achieve acceptable segmentation for a particular type of image using a specific color space. However, it is difficult to find a color space suitable for segmenting different types of color images. This problem can be avoided with the ESNN method we propose here. By feature encoding, a suitable feature vector can be determined automatically by the algorithm, which makes the segmentation adaptable to varied types of color images. To compare our ESNN-based method with the traditional FCM method, we conducted experiments with the results presented in Figs. 10–14. As shown in Fig. 10(c), the red beans in image "BEANS" are distinguished by our method, whereas they are mixed with green beans when suing the FCM method. In Fig. 11(b), the use of FCM does not produce distinguishable result between green, black, blue-green, and brown regions.

TABLE I
CONTRIBUTIONS OF FEATURES TO THE CLUSTERING OF IMAGE "SQUARE3"

Feature	A1	A2	A3	A4	A5	A6	A_	D_f	D_coef
f1	33.51	15.61	41.67	108.83	212.64	322.38	122.44	12356.99	0.9190
f2	153.04	220.19	88.75	119.38	175.70	352.72	184.96	7340.99	0.5460
f3	154.99	221.76	68.96	93.69	54.30	87.57	113.54	3332.65	0.2479
f4	221.02	313.38	134.94	190.91	293.34	488.36	273.66	12820.23	0.9534
f5	0.79	0.79	1.25	1.47	1.51	1.34	1.19	0.09	0.0000
f6	1.36	1.53	0.63	0.85	0.63	0.83	0.97	0.12	0.0000
f7	117.52	103.51	22.81	11.98	126.41	154.64	89.48	2840.66	0.2113
f8	-71.87	-82.93	9.71	2.70	59.34	30.06	-8.83	2683.40	0.1996
f9	-24.61	-29.27	-4.33	-4.05	-27.26	-43.85	-22.23	199.77	0.0149
f10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0000
f11	154.99	145.36	4.72	2.20	14.56	49.43	61.88	4143.50	0.3082
f12	15.61	41.67	108.83	212.64	322.38	115.24	136.06	10862.05	0.8078
f13	220.19	88.75	119.38	175.70	352.72	124.96	180.28	7749.27	0.5763
f14	221.76	68.96	93.69	54.30	87.57	84.84	101.85	3046.90	0.2266
f15	313.38	134.94	190.91	293.34	488.36	192.01	268.82	13446.20	1.0000

TABLE II
CALCULATION OF D_coef FOR EACH COLOR IMAGE

images	beans	clothes	beach	door	S_berry	flower	sample	square2	square3	house
f1	0.6447	0.3508	0.5567	1.0000	1.0000	1.0000	0.3107	0.5671	0.919	1.0000
f2	1.0000	0.3375	0.3372	0.8416	0.6447	0.7052	0.2971	0.6461	0.546	0.2385
f3	0.1253	0.3353	0.2941	0.3297	0.5001	0.595	0.2098	0.1429	0.2479	0.0836
f4	0.2984	1.0000	1.0000	0.3742	0.1253	0.3383	0.2629	1.0000	0.9534	0.891
f5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f7	0.1373	0.0827	0.7836	0.3652	0.2984	0.3700	0.3643	0.5102	0.2113	0.598
f8	0.0475	0.0025	0.0488	0.0004	0.3261	0.1144	1.0000	0.0031	0.1996	0.4509
f9	0.0144	0.0002	0.0002	0.0001	0.0475	0.0232	0.0548	0.0264	0.0149	0.0181
f10	0.0000	0.0000	0.0000	0.0000	0.0144	0.0174	0.0000	0.0000	0.0000	0.6713
f11	0.1224	0.0822	0.7704	0.3656	0.0000	0.2988	0.1143	0.3795	0.3082	0.2723
f12	0.5001	0.2329	0.5235	0.3041	0.1224	0.0877	0.4105	0.3502	0.8078	0.0035
f13	0.3743	0.2383	0.2135	0.2458	0.3743	0.3142	-	-	0.5763	0.024
f14	0.1675	0.2627	0.1253	0.2965	0.1675	0.4324	-	-	0.2266	0.0262
f15	0.3261	0.7100	0.6829	0.2989	0.1373	0.2584	-	-	1.0000	0.0205

TABLE III
PROBABILITIES OF ERRORS IN SEGMENTING IMAGE

images	beans	clothes	beach	door	S_berry	flower	sample	square2	square3	house
RGB	0.2112	0.1800	0.2690	0.3075	0.1558	0.2918	0.0582	0.0611	0.2608	0.1226
SCT	0.2613	0.1930	0.3461	0.3316	0.4020	0.4798	0.3726	0.5949	0.2004	0.1752
YUV	0.4765	0.1984	0.3894	0.3129	0.1113	0.7856	0.0592	0.2718	0.1016	0.1497
HIS	0.4381	0.2172	0.3801	0.3718	0.1368	0.3623	0.4422	0.0646	0.5929	0.2946
L*a*b*	0.3465	0.1818	0.4593	0.3164	0.2882	0.3305	0.3328	0.0706	0.4515	0.2883
ESNN	0.1286	0.1377	0.3008	0.2897	0.1798	0.2918	0.0569	0.0606	0.1323	0.1229

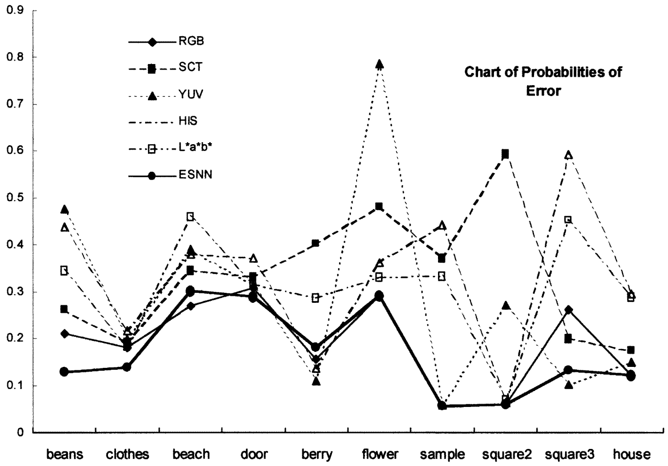


Fig. 15. Probabilities of errors using FCM method in different color spaces and our ESNN method.

Although our method produced some minor faults in brown region in image "SQUARE3," the final result is acceptable. The results in Fig. 12 appear similar. However, in 12(b), three features were used in the segmentation by the traditional method, whereas in 12(c), only two features were needed by our approach. In Fig. 13, the segmentation of image "HOUSE" using FCM method in RGB space is slightly better than that using our method. In Fig. 14(b), part of the shoes is clustered to trousers, but in general, the results by both methods in (b) and (c) are satisfactory. On the whole, the final results using the encoded feature vectors are better than those obtained using a traditional clustering (FCM) algorithm when segmenting the same image. This can be attributed to the way the features are selected in our EFFCM algorithm. With a traditional clustering algorithm, the features used are limited to a particular color space. Problems will then occur if this color space is not properly chosen or it is insufficient for describing the image in concern. Such problems can be prevented by the feature encoding in our ESNN-based approach, which permits optimal selection of the features and feature sequences from different color spaces that best suit the image being processed. Our ESNN-based method offers its advantage over the traditional methods in providing automated feature selection and overcoming the limit by individual color spaces.

The average classification speed of our ESNN-based method is higher than that of the clasical FCM method. Our experiments were performed using MATLAB and C++ programming on a Pentium III 450 platform with 128 MB memory. In a typical segmentation task for a color image, the computation time using our ESNN-based method is around 5–6 min as compared with about 10–15 min by the FCM method. The automated feature selection by our method contributes to the enhanced speed performance in the segmentation.

IV. PERFORMANCE OF EVALUATION

In order to quantitatively evaluate the performance of our ESNN-based method in the image segmentation, the probability of errors between an ideally segmented image and an algorithmically segmented image is adopted [18]. For a two-class image, the probability of error is defined as

$$P(\text{error}) = P(O) \times P(B/O) + P(B) \times P(O/B) \qquad (13)$$

where $P(B/O)$ is the probability (of error) of a region of an object miscounted to the background, and $P(O/B)$ is the probability (of error) of a region of the background miscounted to the object. $P(O)$ and $P(B)$ are the prior probabilities of the object and background, respectively, and are calculated from the ideally segmented image. For a multiclass image, the total probability of error is then given by

$$P(\text{error}) = \sum_{j=1}^{c} \sum_{\substack{i=1 \\ i \neq j}}^{c} P(R_i/R_j) \cdot P(R_j) \qquad (14)$$

where R_i and R_j are the ith and jth regions in an image consisting totally of c regions. For good segmentation, $P(B/O)$, $P(O/B)$, and $P(\text{error})$ should be very small.

Table III lists the probabilities of errors in segmenting each image using the FCM method in different spaces and our ESNN-based method. The probabilities of errors based on ESNN method for image "BEANS," "DOOR," and "CLOTHES," are the smallest. For image "SQUARE2," although the errors by FCM method in RGB, HIS, and CIEL$*$a$*$b$*$ spaces are similar to the ESNN-based method's, exceptionally large errors (59% and 27%) occurred when in SCT and YUV spaces when using FCM. Similar cases occurred with image "SAMPLE," "HOUSE," and "FLOWER," where in RGB space FCM method gives an error rate of about 29%, which is similar to that of the ESNN-based method. However, in other spaces (SCT and YUV spaces), the results of FCM method are highly undesirable with error rates exceeding 48%. Although in some spaces, e.g., RGB for image "BEACH" and YUV for image "STRAWBERRY" and "SQUARE3," the FCM method performs equally well and may even be slightly better than the ESNN-based method; in other spaces, however, it performs much worse with a very high probability of errors. For example, the probability of error is nearly 60% in HIS for image "SQUARE3," 40% in SCT for image "STRAWBERRY," and 46% in CIEL*a*b* for image "BEACH."

The probabilities of errors in the above examples in segmenting the color images by ESNN-based method and FCM method in the five color spaces are plotted in Fig. 15. The curve of "ESNN" turned out to be the best among all the curves. This indicates that the ESNN-based method works well in segmenting all these images and its performance is more stable than that using FCM algorithm in any color space. The very high probabilities of errors of FCM method in some color spaces suggests that it would be highly risky to rely on a particular color space in segmenting different images. The ESNN-based method provides an attractive alternative in this respect.

From the above studies, we observe that the performance of the ESNN-based method in general is better than that of the FCM method in the segmentation. Although, in some particular space, the FCM method may perform slightly better than the ESNN-based method for a specific image, for other images, it performs much worse in the same space. It is seen, for example, while FCM in YUV space performs best in segmenting image "STRAWBERRY," it performs much worse than our ESNN-based method in segmenting image "FLOWER." For general applications, it would be extremely difficult if not impossible to identify a color space that is universally suitable for segmenting all types of images with a traditional segmentation method. On the other hand, with our ESNN-based method, the feature vector suitable for segmenting any input image can be extracted automatically by the algorithm.

V. Conclusions

In this paper, we proposed and implemented a new method for color image segmentation using clustering. With a classical clustering method, although feature vectors from some color spaces may be suitable for segmenting a particular type of color images, it is extremely difficult if not impossible to identify a color space that is universally suitable for segmenting all types of images. This problem can be solved by the proposed ESNN-based method using our feature encoding approach. With the SOFM, a feature vector suitable for segmenting the input image can be extracted automatically by the algorithm. Consequently, the ESNN-based method is an adaptive approach capable of segmenting different types of color images. Without the constraint of a particular color space, the encoded feature vectors are obtained based on the analysis of the contributions of all features to the segmentation. This promises to optimize the segmentation process and save the efforts in manually selecting suitable feature spaces for segmentation tasks. With the selective way the feature vector is formed, our method offers the potential for computation efficiency improvement to general segmentation tasks when a large number of feature vectors are otherwise to be used indiscriminately.

References

[1] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," in *CVGIP*, 1985, 29, pp. 100–132.

[2] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognit.*, vol. 26, pp. 1277–1294, 1993.

[3] E. Littmann and H. Ritter, "Adaptive color segmentation-A comparison of neural and statistical methods," *IEEE Trans. Neural Networks*, vol. 8, pp. 175–185, Jan. 1997.

[4] P. Campadelli, D. Medici, and R. Schettini, "Color image segmentation using Hopfield networks," *Image Vis. Comput.*, vol. 15, pp. 161–166, 1997.

[5] C. L. Huang, "Parallel image segmentation using modified Hopfield model," *Pattern Recogn. Lett.*, vol. 13, pp. 345–353, 1993.

[6] N. Kehtarnavaz, J. Monaco, J. Nimtschek, and A. Weeks, "Color image segmentation using multi-scale clustering," *Proc. IEEE*, vol. 86, pp. 142–147, Jan. 1998.

[7] D. K. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 939–954, Oct. 1995.

[8] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Graph. Image Process.*, vol. 13, pp. 222–241, 1980.

[9] A. Verikas, K. Malmqvist, and L. Bergman, "Color image segmentation by modular neural network," *Pattern Recogn. Lett.*, vol. 18, pp. 173–185, 1997.

[10] J. Gauch and C. W. Hsia, "A comparison of three color image segmentation algorithms in four color spaces," *SPIE Vis. Commun. Image Process.*, vol. 1818, pp. 1168–1181, 1992.

[11] J. Liu and Y. Yang, "Multiresolution color image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 689–700, July 1994.

[12] R. Neviata, "A color edge detection and its use in scene segmentation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 820–826, Nov. 1977.

[13] R. Ohlander, "Analysis of nature scenes," Ph.D. dissertation, Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 1975.

[14] J. Kender, "Saturation, hue and normalized color: Calculation, digitization effects, and use," Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, Tech. Rep., 1976.

[15] L. Lucchese and S. K. Mitra, "An algorithm for unsupervised color image segmentation," Univ. Calif., Santa Barbara, Tech. Rep., 1999.

[16] M. Celenk, "A color clustering technique for image segmentation," in *Comput. Graphics Vis. Image Process.*, vol. 52, 1990, pp. 145–170.

[17] A. Weeks and G. Hague, "Color segmentation in the HIS space using the K-means algorithm," in *Proc. SPIE Symp. Electron. Imag.*, San Jose, CA, Feb. 1997.

[18] B. Cramariuc, M. Gabbouj, and J. Astola, "Clustering based region growing algorithm for color image segmentation," in *Proc. Conf. IEEE Digital Signal Process.*, 1997, pp. 857–860.

[19] H. Cheng and Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," *IEEE Trans. Image Processing*, vol. 9, pp. 2071–2082, Dec. 2000.

[20] N. Li and Y. F. Li, "Method for image segmentation based on ESNN method and its applications," *Opt. Eng.*, vol. 38, no. 5, pp. 908–918, 1999.

Ning Li received the B.S. degree in automatic control from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1990 and the M.Phil. degree in manufacturing engineering and engineering management from City University of Hong Kong in 1999.

She is currently an associate professor with the Department of Electronic Engineering at NUAA. Her research interests include digital image processing, pattern recognition, neural networks, and computer vision.

Y. F. Li (SM'01) received the B.S. and M.S. degrees in electrical engineering from Harbin Institute of Technology, Harbin, China. He received the Ph.D. degree in engineering science from the University of Oxford, Oxford, U.K., in 1993.

From 1989 to 1993, he was with the Robotics Research Group, Department of Engineering Science, the University of Oxford. From 1993 to 1995, he was a postdoctoral research associate with the Artificial Intelligence and Robotics Research Group, Department of Computer Science, University of Wales, Aberystwyth, UK. He is currently an associate professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong. His research interests include robot vision, robot sensing, and sensor-based control.