Pergamon

Neural Networks, Vol. 8, No. 6, pp. 975–987, 1995

0893–6080/95 $9.50+.00

0893-6080(95)00013-5

CONTRIBUTED ARTICLE

A Combined Neural Network Approach for Texture Classification

P. P. RAGHU, R. POONGODI AND B. YEGNANARAYANA

Indian Institute of Technology

(*Received* 6 *January* 1994; *revised and accepted* 24 *November* 1994)

Abstract—*In this article, we present a two-stage neural network structure that combines the characteristics of self-organizing map (SOM) and multilayer perceptron (MLP) for the problem of texture classification. The texture features are extracted using a multichannel approach. The channels comprise of a set of Gabor filters having different sizes, orientations, and frequencies to constitute N-dimensional feature vectors. SOM acts as a clustering mechanism to map these N-dimensional feature vectors onto its M-dimensional output space, where in our experiments, the value of M was taken as two. This, in turn, forms the feature space from which the features are fed into an MLP for training and subsequent classification. It is shown that the disadvantage of using Gabor filters in texture analysis, namely, the higher dimensionality of the Gaborian feature space, is overcome by the reduction in the dimensionality of the feature space achieved by SOM. This results in a significant reduction in the learning time of MLP and hence the overall classification time. It is found that this mechanism increases the interclass distance (average distance among the vectors of different classes) and at the same time decreases the intraclass distance (average distance among the vectors of the same class) in the feature space, thereby reducing the complexity of classification. Experiments were performed on images containing tiles of natural textures as well as image data from remote sensing.*

Keywords—Texture, Texture classification, Multichannel filtering, Multiresolution, Gabor filters, Self-organizing maps, Multilayer, Remote sensing.

1. INTRODUCTION

The problem of texture-based segmentation and classification of images is of considerable interest in many image processing applications. Texture-based image segmentation techniques have been fruitfully used in the analysis and interpretation of radiographic images in medicine, seismic trace images, and earth cover images obtained using remote sensing techniques (Berger, 1970; Weszka, Dyer & Rosenfeld, 1976). Analysis of textures requires the identification of proper attributes or features that differentiate the textures in the image for further segmentation, classification, and recognition. A variety of feature extraction and classification techniques had been suggested in the past for the purpose of texture analysis (Haralick, Shanmugam & Distein, 1973; Haralick, 1979) and more efficient methods have been suggested recently, for example, in Keller, Chen and Crownover (1989). These methods had been utilized by many researchers in different contexts.

In spite of the importance of texture and its presence in many real and synthetic images, it is hard to find a reasonable quantitative definition for texture that is general enough. One definition for texture describes it as a repetition of some primitive patterns in space. But such a definition may be appropriate only while referring to deterministic patterns such as those that can be easily synthesized or those that occur in nature as regular patterns (for example, ripples on desert sands, snake skin, etc.) or those that are man-made (for example, printed circuit board patterns, brick walls, etc.). In contrast, many man-made and natural images (for example pizza surfaces, clouds, ocean surface, etc.) possess only a stochastic structure. Furthermore, it must be noted that the interval between repetitions of the primitive patterns

Acknowledgements: This work was supported by Indian Space Research Organization. We thank Prof. B. L. Deekshatulu and Dr. Jairam Hebbar (National Remote Sensing Agency, Hyderabad) for helpful discussions about the texture analysis in remote sensing and also for providing the remote sensed data for our experiments.

Requests for reprints should be sent to Prof. B. Yegnanarayana, Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India.

need not be constant and can be space-variant in one or more directions. Another explanation of texture describes it to be a spatial relation between intensity values of neighboring pixels, repeated over an area larger than the size of the relation. Thus, the diversity of natural and artificial textures makes it very difficult to give a universal definition of texture. We prefer to adopt the definition suggested by Sklansky (1979), because of its generality: "a region in an image has a constant texture if a set of local statistics or other local properties of the picture are constant, slowly varying, or approximately periodic." This definition explains many of the textures found in natural images like those used in remote sensing and medical imaging applications. The local statistics or properties that are repeated over the textured region can be called texture elements, or texels. It must, however, be noted that texture has both local as well as global meaning—it is characterized by invariance of certain local attributes, as well as possessing properties described over a region of an image. Thus, it is the local attribute of the texture that is used in the identification of a texture type and its global property for the segmentation of the image.

Recently, the resurgence of interest in the artificial neural networks has brought to focus their use in different stages of texture analysis. In contrast to the traditional techniques, neural network-based methods have the following advantages. (1) Any arbitrary functional relationship between the input and output patterns can be captured by a neural network and this relationship need not be known or prescribed explicitly. (2) No assumptions need to be made regarding the statistical distributions underlying the input patterns. (3) Neural networks are fault-tolerant in the sense that even if some of the connections or processing elements are degraded or lost, the performance of the network may not be much affected. The performance of a neural network degrades gracefully as the damages to the network increase.

Some of the previous efforts on the development of neural network-based approaches for texture analysis and texture-based image segmentation can be found in Dupaguntla and Vemuri (1989), Leung and Peterson (1991), and DeCosta and Chouikha (1992). Dupaguntla and Vemuri (1989) suggested a composite architecture consisting of (1) Grossberg's hierarchical network for feature extraction and (2) Kohonen's adaptive learning network for texture feature discrimination. Ari Visa (1990) used gray tone co-occurrence matrix for feature extraction and a self-organizing map for clustering the features in the class-membership space. Several deterministic and stochastic relaxation algorithms implemented using highly parallel networks have also been suggested recently (Chellappa, Manjunath, & Simchony, 1992). Raghu, Chouhan, and Yegnanarayana (1993) described the use of a multilayer perceptron for the classification of multispectral textures from remote sensing using the Gabor features and shown the classification performance to be immune to additive noise. Because conventional methods based on second-order statistics fail to classify textures with identical first- and second-order statistics, an approach based on higher-order statistics in conjunction with a neural classifier was suggested by DeCosta and Chouikha (1992). This procedure also was shown to be robust in the presence of additive Gaussian noise. Greenspan, Goodman, and Chellappa (1991) used a hybrid unsupervised and supervised scheme for texture segmentation that is based on Kohonen's self-organizing map for quantizing the features and a rule-based system for learning and classifying the class code books. Results were provided on natural and synthetic textures.

In this work, we propose a texture classification framework based on a two-stage neural network model comprised of Kohonen's self-organizing map (SOM) and multilayer perceptron (MLP). The texture features are extracted from the image using a bank of Gabor filters. SOM acts as a clustering mechanism that projects N-dimensional features from the filter bank into an M-dimensional feature space. The resulting vectors are fed into an MLP that categorizes them onto one of the prelearned texture classes. The proposed scheme is a good example of how different neural network models can be cascaded to reduce the complexity of classification. A similar neural network approach was used by Huang and Kuh (1992) in another context where they used it for isolated word recognition to obtain high recognition rate. They mapped features from each frame of the word onto the SOM output to form a trajectory of winner nodes for a given word. The MLP learns this trajectory for each word and classifies it.

Various texture analysis techniques based on the spatial filtering approaches were attempted in the past to characterize the textures on the basis of the frequency contents and the orientations of texture elements in the textures (Laws, 1980; Coggins & Jain, 1985). Bajcsy (1973) proposed a Fourier filtering method for texture analysis that makes use of the spatial frequency contents of subimages by windowing the original textured image. This method characterizes the spatial frequency content, but does not consider the information in the spatial domain. The information regarding the spatial frequency alone will not be adequate in many cases, especially when the textures are nonstationary as in the case of remotely sensed images. To take into consideration the spatial as well as spatial frequency characteristics of the textured image, a number of spatial/spatial–frequency (s/sf) analysis methods have been suggested

in the past (for example, see Jacobson & Wechsler, 1988).

In 1946, Gabor proposed a set of narrow band filters that can highly concentrate in time and frequency domains for conjoint time–frequency signal representation (Gabor, 1946). This idea was extended to 2-D by Daugman (1985, 1988) for representing an image into a number of localized frequency channels measured at different resolutions. The use of 2-D Gabor filters is well recognized in the recent past as a joint spatial/spatial–frequency representation for images that can perform multichannel–multiresolution feature extraction for texture segmentation (Turner, 1986; Bovik, Clark, & Geisler, 1987). Gabor filters have an important theoretical property concerned with the optimal localization in the spatial and spatial frequency domains simultaneously.

The remainder of this paper is organized as follows. The next section starts with the description of the feature extraction scheme based on the multichannel Gabor filters. It reviews the properties of the filters that make them suitable for texture analysis and discusses issues in using them. Section 3 details the proposed two-stage neural network, where different components of the architecture are described. In Section 4, experimental results on classifying different types of textures from different images, including remotely sensed imagery, are reported. Finally, in Section 5, a few comments and remarks are given on the current work.

2. MULTICHANNEL–MULTIRESOLUTION FEATURE EXTRACTION SCHEME

The gray level value and the spatial location of a pixel in an image can be considered as first-order features (Coleman & Andrews, 1979). These features extract image details that are not related to context. The higher-order features describe the relationship of the given pixel to its surrounding pixels in the image. In the analysis of image textures, it is not possible to extract the information content directly from the first-order features. This is because texture is a contextual property dealing with local variations of gray level intensity and hence can be described only through a spatial transform on a group of related pixels at particular resolutions defined by the texel sizes. An important step in texture analysis involves identification and extraction of some texture features that can efficiently characterize different textures in an image.

Generally, the image textures to be analyzed have texels of different sizes. So the optimal resolution with which the features are to be extracted to analyze the image textures cannot be defined *a priori*. One efficient way to represent the different image details is to reorganise the image into a number of subsampled approximations of it at different resolutions. This way of image reorganization is called the multiresolution representation (Mallat, 1989a,b). This scheme analyses the coarse image details first and gradually increases the resolution to analyze the finer details. Fuller discussion of multiresolution image analysis can be found in Rosenfield (1982) and Mallat (1989a).

Textured images can have space-varying local properties. The variations in local properties can be due to variations in the features such as the orientation, frequency, and size of the texture elements. Thus, an appropriate mechanism for extraction of these features is essential. The feature extraction is followed by the classification stage where a classification algorithm maps the points in the feature space onto points in the class-membership space. The accuracy of texture classification depends on the complexity in the distribution of the extracted features.

The feature extraction method adapted for our experiments is based on the mechanism of multichannel representation of the retinal images in the biological visual system. Studies on the biological visual system had shown that several visual cortical areas of mammals contain a large number of linear and nonlinear neurons having receptive field profiles with selectivities for a variety of stimulus attributes such as location in the 2-D visual space, orientation, motion, stereoscopic depth, and spatial frequency (Van Essen, 1979). The receptive fields of the simple cells in the early visual system possess some interesting properties that make the visual representation capable of having space-domain local feature extraction confined to spatially oriented frequency channels that are quasi-independent in nature (Campbell & Robson, 1968). Marceljia (1980) proposed that these receptive fields in cross-section could be described by 1-D Gabor elementary functions, and simultaneously Daugman (1980) proposed the 2-D Gabor filter model for such receptive fields. When appropriately tuned, these filters are found to be extremely useful for performing texture feature extraction and texture edge detection. More details regarding the biological visual system and Gabor filter models of cortical cells can be found in De Valois, Albrecht, and Thorell (1982) and Pollen and Ronner (1983).

Recently, several studies on the analysis of textures using Gabor filters have appeared (Bovik, Clark, & Geisler, 1990; Jain & Farrokhnia, 1991). The local properties of a texture can be obtained using a set of Gabor filters with appropriately chosen filter orientation, frequency and size. The 2-D Gabor filters are complex sinusoidal gratings modulated by 2-D Gaussian functions, thus forming the complex

valued functions in R^2. The frequencies and orientations of the complex sinusoid in the Gabor filter describe the local structure of the texture in different channels whereas the Gaussian envelope in the Gabor filter defines the resolution with which the texture structure is characterized. Thus, Gabor filters allow us to obtain information regarding differences in dominant sizes, orientations, and distributions of texture elements in different textures. The general form of 2-D Gabor functions is given by

$$g(x, y, k_x, k_y, \sigma_x, \sigma_y) = Ae^{-1/2[(x/\sigma_x)^2+(y/\sigma_y)^2]}e^{j(k_x x+k_y y)}$$

where A is a scaling factor.

The spatial extent of the Gabor function is defined by (σ_x, σ_y). The orientation of the span-limited sinusoidal grating is given by $\tan^{-1}(k_y/k_x)$ and its frequency is specified along the x and y coordinates by k_x and k_y, respectively. Figure 1 shows a typical Gabor filter.

Gabor filtered output of the image is obtained by the convolution of the image with the Gabor function. The power spectrum of the filtered images at each pixel position is used as features to characterize the pixel. If $I(x, y)$ is the image, then the feature value at position (x, y) of the image is given by

$$f(x, y) = |I(x, y) * g(x, y)|^2$$

for a filter $g(x, y)$ with given k_x, k_y, σ_x, and σ_y, where $*$ denotes convolution operation.

The assumption in using the Gabor filters for texture processing is that, each texture is characterized by a given localized spatial frequency or a narrow range of dominant localized spatial frequencies that differ significantly from dominant frequencies of other textures. Gabor functions possess several desirable properties useful for texture analysis. The minimal time-bandwidth product of the Gaussian function makes the Gabor function concentrate both in spatial and spatial–frequency domains, optimizing the trade-off between spatial feature extraction and spatial frequency selection (Daugman, 1985). This means that the Gabor filter can be defined to have narrow frequency and orientation responses while preserving the spatial localization. Moreover, the Gabor filters represent a complete but nonorthogonal basis, and it is thus possible to find a set of expansion coefficients that reconstructs the image with minimum error between the original and reconstructed image (Daugman, 1988). The textures that have significantly different spatial frequencies can be encoded into multiple channels, each having narrow spatial frequency and orientation. The local information regarding the texture elements is described by the orientation and frequency of the sinusoidal grating, and the local window of frequency analysis is provided by the Gaussian envelope of the Gabor function.

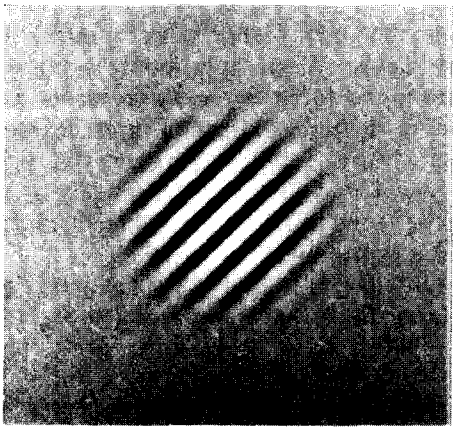

FIGURE 1. A typical Gabor filter. The grating is sinusoid modulated by Gaussian function. This filter has an orientation of 45 degrees.

One common approach for filter selection is to find the frequency components by which different textures in the image can be characterized (Bovik et al., 1990). But textures in natural images are characterized by a band of frequencies rather than a single frequency. Also, it is difficult to find a single resolution with which each texture can be characterized. To capture small changes in the frequency and orientation, the number of filters required will be large. Hence, it is not possible to model natural textures with a single filter for each texture. The solution we have adopted is to use a number of such filters with different bandwidths, frequencies, and orientations so that some of these filters are used to suit each texture. Now the problem turns into an increased dimensionality of the feature vectors extracted from the Gabor filtered image. To deal with this problem, we use a two-stage neural network system discussed in Section 3.

3. THE ARCHITECTURE OF THE PROPOSED CLASSIFICATION SCHEME

The proposed classification scheme is comprised of a hierarchical organization of SOM and MLP. SOM receives inputs from the Gabor filter bank and maps onto an M-dimensional space where M is the dimensionality of the SOM output node distribution. The transformed feature vectors are fed into the MLP, which classifies them. We call the feature space generated from the Gabor filter output as primary feature space and M-dimensional feature space from SOM output as secondary feature space. The vectors from the secondary feature space are called secondary feature vectors. The concept behind the use of SOM as an intermediate stage is that it can perform a

topology-preserving feature mapping from its input space to output space, and these mapped features, which are of reduced dimension, can represent the necessary information in the input features. Thus, the training and classification of the upper stage (MLP) can be done in a reduced dimension compared to the higher dimension of the primary feature space. Each stage is described below in detail.

3.1. Secondary Feature Extraction Stage

This stage uses the topology-preserving self-organizing feature map proposed by Kohonen (1990) to extract the secondary feature vectors from the input data. The feature maps constitute an important class of natural and artificial neural systems (Knudsen, du Lac, & Easterly, 1987). In a topology-preserving map, units located physically next to each other in an M-dimensional output space will respond to the classes of input vectors that are similar in the N-dimensional input space. So the similarity of signals is converted into proximity of the excited neurons. A remarkable property of SOM is that the dimensionality of the sensory input space (N) and of the output neuron space (M) need not agree, and usually M is chosen to be much less than N. While preserving the neighborhood relations between neurons, the best mapping is carried out in such a way that the input space is uniformly covered by associated neurons in the output space. Thus, larger dimension data are projected down to a smaller dimension map in a way that maintains the natural order of the input vectors. To reduce the training time for the classifer (MLP) while preserving the information in the N-dimensional Gabor filter inputs, we have selected a 2-D output space of SOM. Hence, the SOM contains an input layer having N nodes and a 2-D lattice of output nodes.

Kohonen's algorithm is a well-established learning rule in the field of neural networks for training the self-organizing feature maps. The algorithm describes a map $\Phi_W : R^N \rightarrow R^M$ from the input space X onto the output space Y, which assigns to each element $x \in X$, an element $y_j = \Phi_W(x) \in Y$ corresponding to a node j in the output space (Ritter, Martinetz, & Schulten, 1992). The map Φ_W is defined by the synaptic strengths $W = \{\mathbf{w}_r\}$, $\mathbf{w}_r \in \mathrm{R}^{\mathrm{N}}$ for each node r in the output space. For a given input stimulus $\mathbf{x} \in X$, the node j and hence the image $\mathbf{y}_j$, which is a point in the M-dimensional space for the node j, are determined by the following condition,

$$|\mathbf{w}_j - \mathbf{x}| = \min_r |\mathbf{w}_r - \mathbf{x}|$$

Here j is called the winner node for the input stimulus $\mathbf{x}$. Thus, an element $\mathbf{x} \in X$ applied to the input layer is mapped onto an element $\mathbf{y}_j \in Y$, which is the coordinate of the node j in the output layer for which the distance between the element $\mathbf{x}$ and the synaptic weight vector $\mathbf{w}_j$ is minimum. The nodes in the output space are arranged as an M-dimensional lattice. Each node in the output layer is connected from all nodes in the input layer and has lateral interaction with other nodes in the output layer. This lateral interaction is defined by a neighborhood function, $\Lambda_{j,r}(t)$ of the distance between the winner node j and any other node r in the output layer. A typical neighborhood function is a Gaussian function given by

$$\Lambda_{j,r}(t) = e^{-|\mathbf{y}_j - \mathbf{y}_r|^2/2\sigma^2(t)}$$

where $\sigma(t)$ is the width of the Gaussian function that decreases as the number of training epochs increases.

The learning in SOM slowly establishes the map Φ_W by incremental adjustments in each connection weight $\mathbf{w}_r$. The following weight update rule given by Kohonen is used to train the network,

$$\mathbf{w}_r(t+1) = \mathbf{w}_r(t) + \Delta\mathbf{w}_r(t)$$

where,

$$\Delta\mathbf{w}_r(t) = \eta\Lambda_{j,r}(t)(\mathbf{x} - \mathbf{w}_r(t))$$

The SOM clusters the feature vectors from the Gabor filter bank to achieve a compact representation, but with most of the information preserved. The network is trained to cluster the input patterns in the output space. The number of clusters will be equal to the number of classes in the input patterns. After training, the feature vector from the Gabor filter bank is fed to SOM, which makes one of the nodes in the output layer to win. The normalized coordinates of this winner node in the output layer forms the M-dimensional (in our experiments, M = 2) secondary feature vector to train the MLP classifier. The normalization involves dividing each component of the coordinate by the total number of nodes in that direction.

The general transformation from the image space to the feature space need not be recoverable, and it depends on the application. For the requirements like image compression, the transformation should be recoverable; violating this will result in information loss. But applications like classification and segmentation require only a minimal set of features that will distinguish different textures, and the transformation need not be recoverable. The transformation from image to the Gaborian space is recoverable to get back the original image, hence the Gabor filters can completely represent a given image. The secondary

features from SOM are not recoverable, but they preserve the information necessary to discriminate different classes in the input feature space.

3.2. Classification Using Multilayer Perceptron

If the data to be classified are noise-free and unambiguous, then the SOM itself is sufficient for classification because the clusters formed in its output space will have crisp boundaries and neurons belonging to the clusters can be assigned with the corresponding class labels. But this cannot be assured for natural data because the cluster boundaries could be fuzzy. The dispersion of the cluster(s) of a given class is dependent on the dispersion of the input patterns for the same class in the N-dimensional feature space. For the purpose of learning the cluster

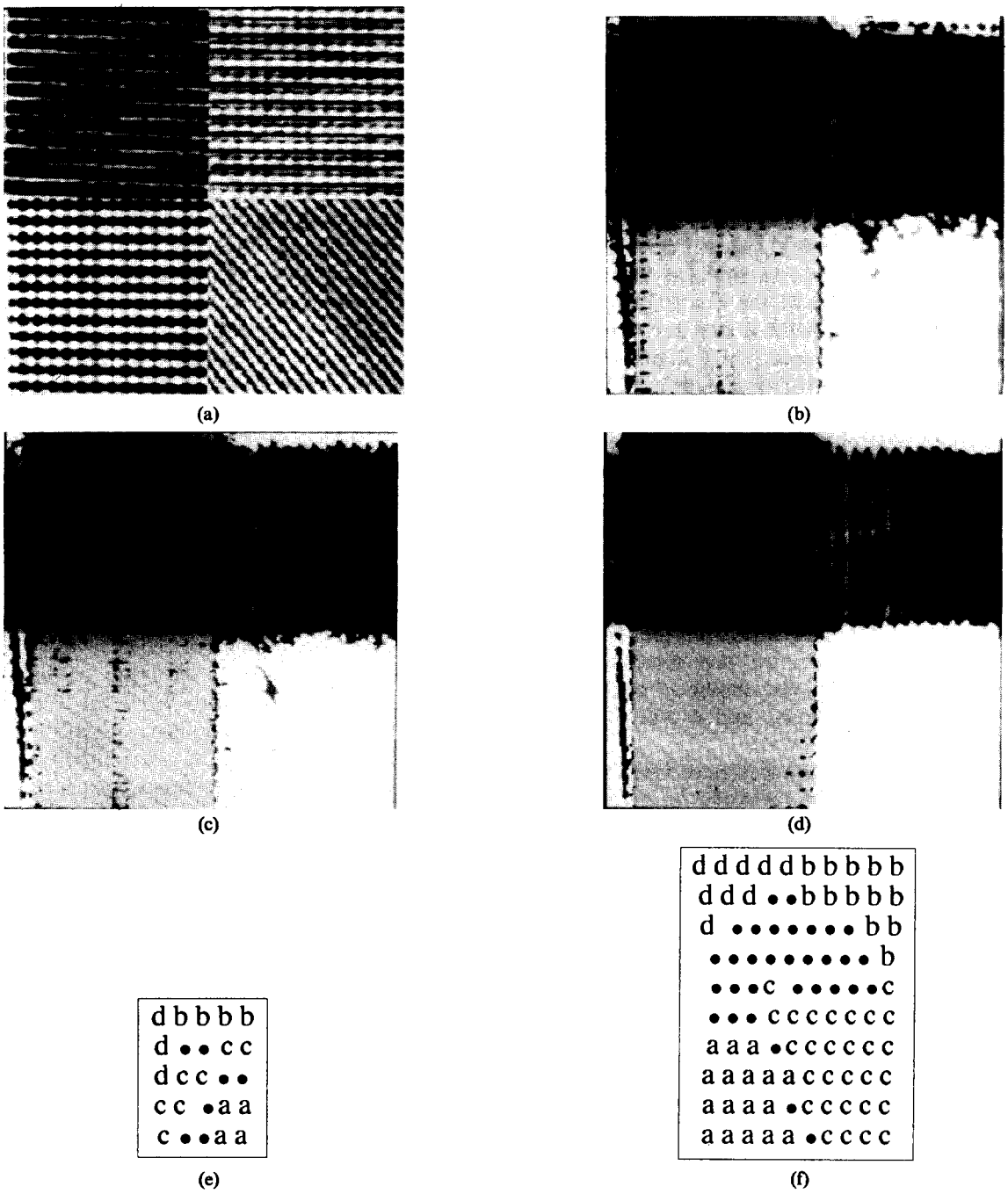

FIGURE 2. The classification of textures: Textured image-1. (a) Original textured image. (b) Classified image with 5 × 5 SOM output array and MLP. (c) Classified image with 10 × 10 SOM output array and MLP. (d) Classification using MLP alone. (e) Cluster map of SOM output layer: 5 × 5 case. (f) Cluster map of SOM output layer: 10 × 10 case. Four classes.

distributions, we have used a two-layer feedforward neural network. The texture class labels are assumed to be known *a priori*, and hence error back-propagation algorithm (Rumelhart, Hinton, & Williams, 1986) was used for training the network.

The working of this network hierarchy consists of two phases—training phase and classification phase. Known regions from the image are used as training sites to train the network. The texture features at each pixel in the imagery is extracted using the filter bank and applied to SOM. The competition in the SOM output layer makes one node to win in response to the input. The normalized coordinates of the winner node of SOM are then fed to the MLP. The training phase contains two steps. First, SOM is trained with Kohonen's unsupervised training algorithm using the features from the Gabor filter. The SOM training is followed by training of the MLP. During the image segmentation and classification phase, the feature vectors corresponding to each pixel in the image is given a class label at the output of the MLP. If the MLP outputs are not distinguishable, or if they are below a particular threshold, the pixel is labeled as unclassified.

There are a number of advantages of the proposed scheme compared to the classification scheme where the feature vectors from the filter bank are given directly to an MLP network. One main advantage is the dimensionality reduction of the feature space. This reduces the training time of the classifier significantly. Further, it was observed that the interclass separation was increased and the intraclass separation was decreased for the secondary feature vectors from SOM than for the primary feature vectors from the filter bank. This is demonstrated in the results of classifying different images. But the efficiency and performance of the classification depends on the parameters of SOM, especially the neighborhood dependency of the winning node and the number of training epochs.

4. RESULTS AND DISCUSSION

Performance of the proposed scheme was analyzed using a variety of textured images, including remotely sensed imagery. The images were categorized in increasing order of difficulty. The first set of images consists of texture tiles made up of natural textures and is characterized by known texture boundaries and unknown texture models. The second category consists of images of natural scenes, where processing becomes difficult due to both unknown boundary and unknown texture model. The classifier used in all the experiments is an MLP with one hidden layer. In each case, the results obtained using the proposed method as well as the result when the filter bank outputs are directly fed to MLP without SOM interface are shown. The normalized distance matrices of the primary feature vectors from the filter bank and of the secondary feature vectors from SOM are also shown. Each result is provided with a SOM cluster map that is formed by giving class labels (a, b, c . . .) to the winner nodes when the input patterns of known classes are applied to SOM.

Figure 2a shows an image of size 256 × 256 and containing four textures, all are of near-deterministic nature with known texture boundaries. We have used 36 Gabor filters, having three different bandwidths ($\sigma_x = \sigma_y = 6.25$, 12.5, and 25), three frequencies (corresponding to wavelengths of 2.5π, 5π, and 10π pixels/cycle) and four orientations (0, 45, 90, and 135 degrees), constituting a 36-dimensional feature vector representing each pixel in the image. We used features from 600 pixels from the center of each texture for training. Table 1a shows the distance matrix of the primary feature vectors of the training samples, and it shows the interclass as well as intraclass distances. Figure 2b shows the classified image when the SOM has an output node array of 5 × 5. The corresponding distance matrix of the secondary feature vectors from the SOM is shown in Table 1b. The cluster map in the SOM output layer is shown in Figure 2e. This shows four distinct clusters, each one for one class. Figure 2c shows the classification result when a SOM with 10 × 10 output nodes was used. The corresponding secondary features distance matrix is shown in Table 1c. The cluster map in this case is given in Figure 2f. The classification when the Gabor features are fed directly to MLP without SOM is shown in Figure 2d. The results show that the classification performance has improved using the proposed scheme.

Figure 3a shows a 256 × 256 point image contain-

TABLE 1
Distance Matrices for Image shown in Figure 2a

	Texture 1	Texture 2	Texture 3	Texture 4
(a) Distance matrix of raw feature vectors from the Gabor filters.				
Texture 1	0.6036	0.9517	0.9832	1.0000
Texture 2	0.9517	0.4481	0.7020	0.5969
Texture 3	0.9832	0.7020	0.6096	0.8147
Texture 4	1.0000	0.5969	0.8147	0.2728
(b) Distance matrix of secondary feature vectors from SOM output: 5 × 5 case				
Texture 1	0.0031	1.0000	0.7446	0.8023
Texture 2	1.0000	0.2162	0.6400	0.6697
Texture 3	0.7446	0.6400	0.3240	0.8340
Texture 4	0.8023	0.6697	0.8340	0.2308
(c) Distance matrix of secondary feature vectors from SOM output: 10 × 10 case				
Texture 1	0.1133	1.0000	0.6878	0.7451
Texture 2	1.0000	0.1420	0.6978	0.6960
Texture 3	0.6878	0.6978	0.1705	0.9573
Texture 4	0.7451	0.6960	0.9753	0.1050

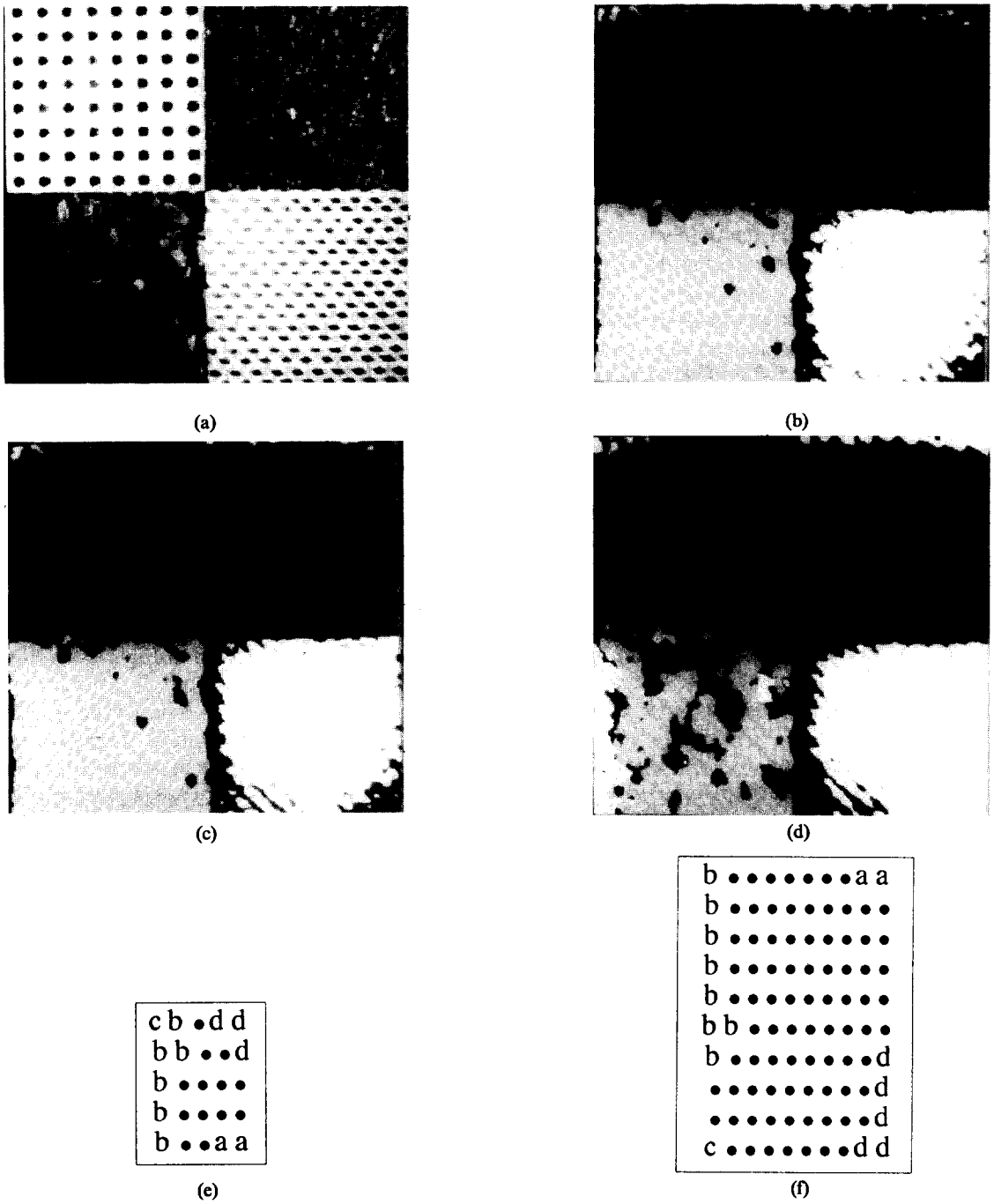

FIGURE 3. The classification of textures: Textured image-2. (a) Original textured image. (b) Classified image with 5 × 5 SOM output array and MLP. (c) Classified image with 10 × 10 SOM output array and MLP. (d) Classification using MLP alone. (e) Cluster map of SOM output layer: 5 × 5 case. (f) Cluster map of SOM output layer: 10 × 10 case. Four classes.

ing four textures, two of them are nearly deterministic (dots and diamonds) whereas the other two are of stochastic nature (sand and pebbles). In this case also, texture boundaries are known. We have used 36 Gabor filters, having three different bandwidths, three frequencies, and four orientations (0, 45, 90, and 135 degrees), constituting a 36-dimensional feature vector representing each pixel in the image. The chosen bandwidths and frequencies are the same as those in the previous experiment. Here, we have used features from 400 pixels selected from the center of each texture for training the network. Table 2a shows the distance matrix of the primary feature vectors of the training samples. Figure 3b shows the classified image using a SOM with an output node array of 5 × 5. The distance matrix of the secondary feature vectors is shown in Table 2b, and the cluster map of the SOM output is shown in Figure 3e. Figure

TABLE 2
Distance Matrices for Image shown in Figure 3a

	Texture 1	Texture 2	Texture 3	Texture 4
(a) Distance matrix of raw feature vectors from the Gabor filters				
Texture 1	0.3307	0.7296	1.0000	0.6300
Texture 2	0.7296	0.1513	0.3482	0.3549
Texture 3	1.0000	0.3482	0.1403	0.5695
Texture 4	0.6300	0.3549	0.5695	0.1394
(b) Distance matrix of secondary feature vectors from SOM output: 5 × 5 case				
Texture 1	0.0000	0.7700	1.0000	0.6848
Texture 2	0.7700	0.1812	0.4286	0.8158
Texture 3	1.0000	0.4826	0.0000	0.7081
Texture 4	0.6848	0.8158	0.7081	0.0426
(c) Distance matrix of secondary feature vectors from SOM output: 10 × 10 case				
Texture 1	0.0011	0.7390	1.0000	0.6895
Texture 2	0.7390	0.1467	0.5065	0.8436
Texture 3	1.0000	0.5065	0.0000	0.6930
Texture 4	0.6894	0.8436	0.6930	0.0523

3c shows the classification result when a 10 × 10 array of output nodes was used, and the corresponding secondary features distance matrix is shown in Table 2c. The cluster map in this case is given in Figure 3f. The classification result of using MLP alone is shown in Figure 3d. The results of the above two experiments show that the proposed scheme gives much better performance compared to the scheme where the MLP alone is used. The distance matrices in each case give the intraclass distances (the diagonal elements in the matrix) and the interclass distances (elements other than the diagonal). The diagonal elements in the distance matrices produced from SOM outputs are significantly lower compared to the diagonal elements of the distance matrices of the primary feature vectors.

The next two images belong to the second category described above. They are characterized by unknown texture models as well as unknown boundaries. They are samples from remotely sensed imagery. Figure 4a shows the image of a particular region in the Venus surface, taken by the Magellan spacecraft. The image is of size 256 × 256, and it contains two textures. In this case, we have used 12 Gabor filters, having single bandwidth ($\sigma_x = \sigma_y = 6.25$), three frequencies ($2.5\pi$, and 10π pixels/cycle wavelengths) and four orientations (0, 45, 90, and 135 degrees), constituting a 12-dimensional feature vector representing each pixel in the image. Training is done using the features from 200 pixels of each texture. Table 3a shows the distance matrix of the primary feature vectors of the training samples. Figure 4b shows the classification result when a SOM with an output node array of 5 × 5 was used. The distance matrix of the secondary feature vectors is shown in Table 4b, and the cluster map from the SOM output is shown in Figure 4e. Figure 4c shows the classification with the SOM having 10 × 10 output nodes. The distance matrix of secondary features is shown in Table 3c, and the cluster map is given in Figure 4f. The classification result with MLP alone is shown in Figure 4d. The performance of the proposed method for this image is similar to that when MLP alone was used. But the advantage here is in the time taken for the classification, which is given in Table 4 and explained later in this section.

Figure 5a shows an IRS remote sensed image. The image is of size 256 × 256. We selected four textures for supervised classification. Here also, 36 Gabor filters, having three different bandwidths, three frequencies, and four orientations (0, 45, 90, and 135 degrees) are used, constituting a 36-dimensional feature vector representing each pixel in the image. In this case also the filter parameters are the same as in the first experiment. Figure 5b shows the classified image with a SOM output node array of 10 × 10. The cluster map of the SOM output is shown in Figure 5d. The classification result by using the MLP alone is shown in Figure 5c. The results show a significantly better performance of the proposed method compared to the case where Gabor features are fed directly to the MLP, where the classes are not well distinguished. The cluster map in this case (Figure 5d) shows more than one cluster for each class except for the class *d*. So it is not useful to compute distance matrix for the secondary feature vectors. Also, a few nodes get activated for more than one class, thereby making the cluster boundaries fuzzy. The purpose of the MLP here is to learn the complex cluster distribution and also to defuzzify the cluster boundaries.

Another important aspect is the overall time taken for training and classification. The time for each case when the experiments were conducted on an i860-

TABLE 3
Distance Matrices for Image shown in Figure 4a

	Texture 1	Texture 2
(a) Distance matrix of raw feature vectors from the Gabor filters		
Texture 1	0.1650	1.0000
Texture 2	1.0000	0.1013
(b) Distance matrix of secondary feature vectors from SOM output: 5 × 5 case		
Texture 1	0.0104	1.0000
Texture 2	1.0000	0.0233
(c) Distance matrix of secondary feature vectors from SOM output: 10 × 10 case		
Texture 1	0.0485	1.0000
Texture 2	1.0000	0.0922

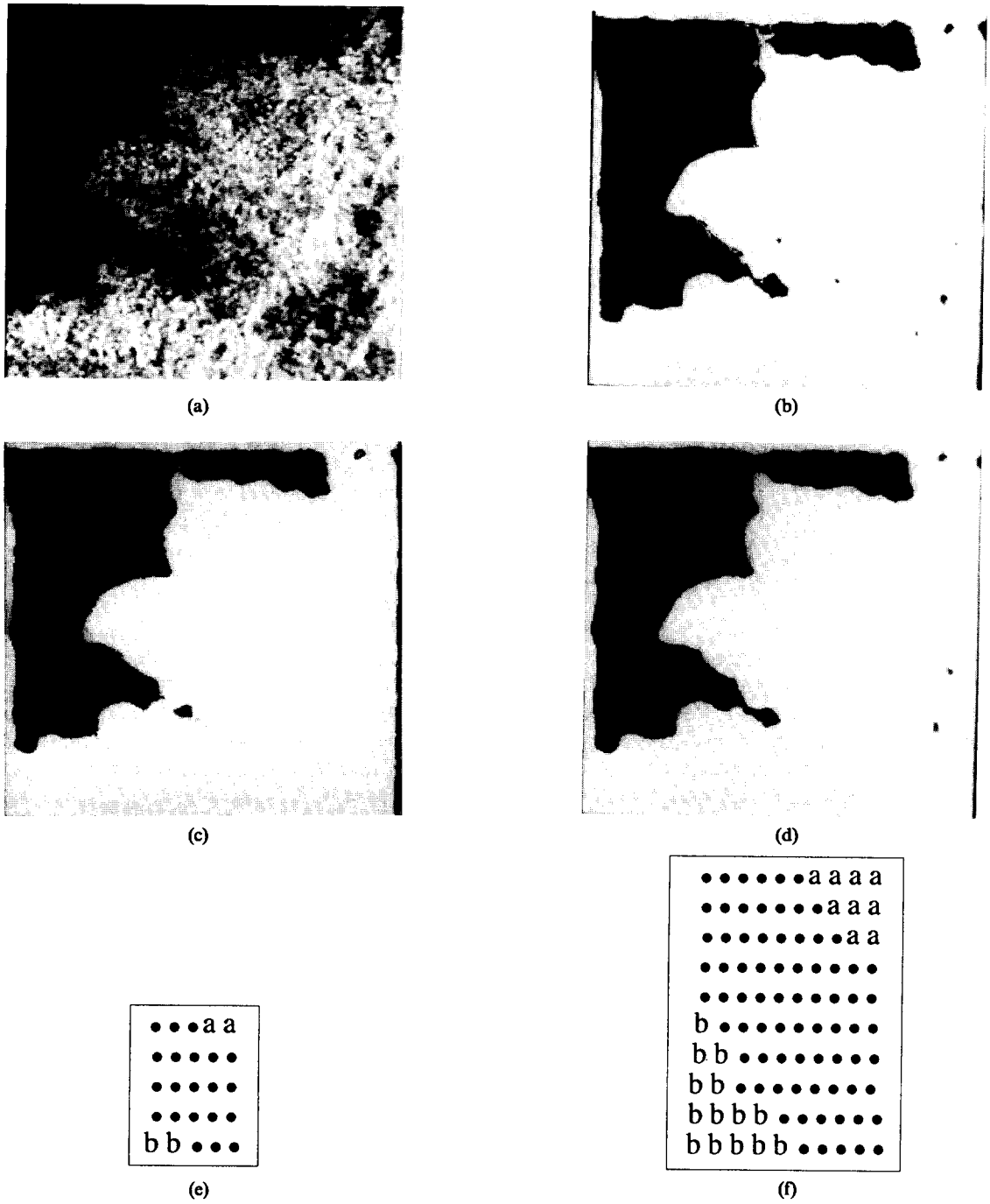

FIGURE 4. The classification of textures: image of Venus surface. (a) Original textured image. (b) Classified image with 5 × 5 SOM output array and MLP. (c) Classified image with 10 × 10 SOM output array and MLP. (d) Classification using MLP alone. (e) Cluster map of SOM output layer: 5 × 5 case. (f) Cluster map of SOM output layer: 10 × 10 case. Two classes.

based UNIX workstation is shown in Table 4. It shows real time, user time, and system time separately for each case. It can be seen that the time taken by the two-stage method is much lower with the same or improved performance in classification.

5. CONCLUSION

In this paper, we have proposed a texture classification scheme for the classification and segmentation of textured images based on a two-stage neural network approach. The scheme uses Gabor filters of different bandwidths, orientations, and frequencies for feature extraction. The classification phase is achieved by a self-organizing map cascaded with a two-layer feedforward neural network trained using the back-propagation algorithm. The performance of the system in classifying a number of textured images was studied and the performance was compared with a scheme that uses only a two-layer network for

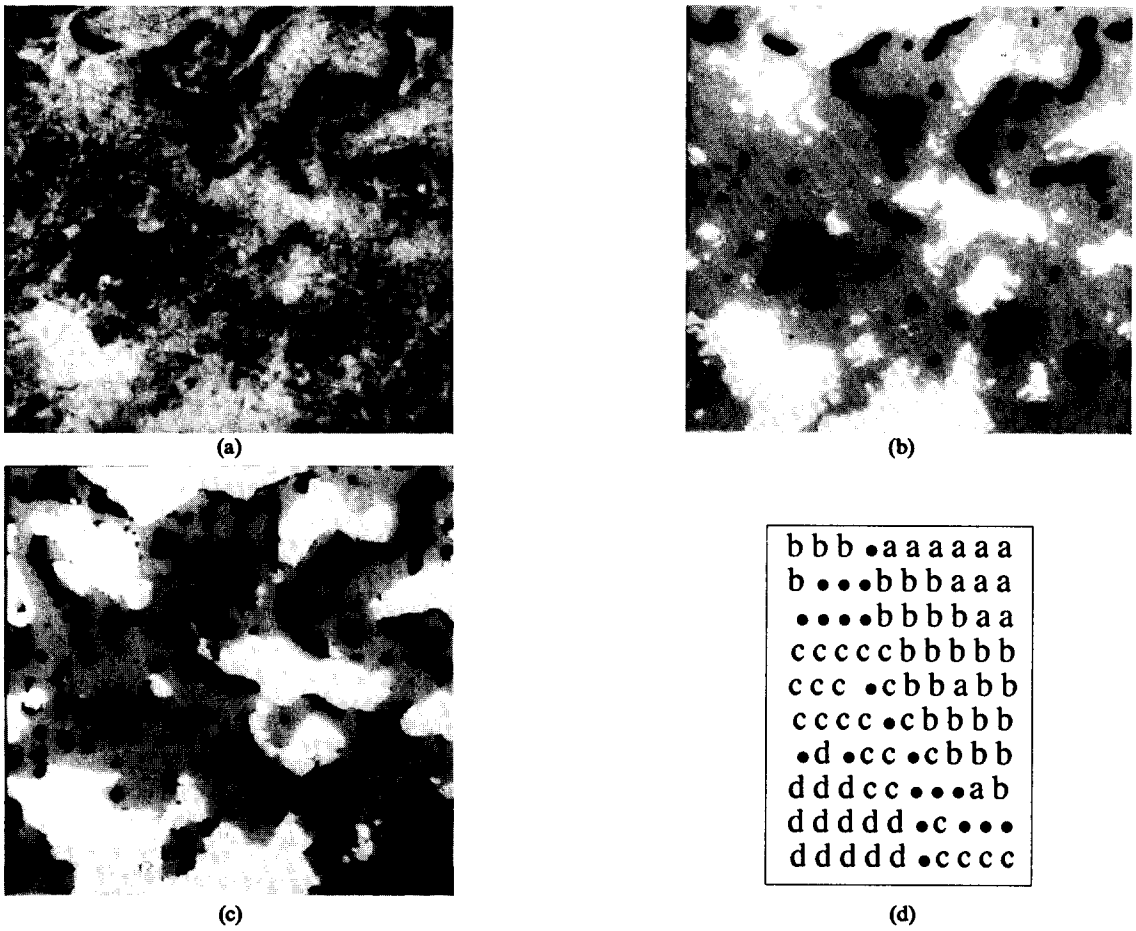

FIGURE 5. The classification of textures: image from IRS data. (a) Original textured image. (b) Classified image with 10 × 10 SOM output array and MLP. (c) Classification using MLP alone. (d) Cluster map of SOM output layer. 10 × 10 case. Four classes.

classification. It is found that the incorporation of secondary feature extraction using SOM improves the classification performance and reduces the complexity of the feature extraction and classification stages. Finally, the required time for classification based on the proposed scheme is much less than that for the classification using only MLP.

It must be noted that the performance of the

TABLE 4
The Time Taken by Different Classification Experiments Conducted on Godrej I860-based Workstation

Name of Image	Type*	Time for MLP alone (H : *min* : *s*)	Time for SOM + MLP (5 × 5 SOM Output) (H : *min* : *s*)	Time for SOM + MLP (10 × 10 SOM Output) (H : *min* : *s*)
Textured Image 1	real	3 : 34 : 35.1	1 : 23 : 41.5	1 : 28 : 22.1
(Figure 2a)	user	1 : 36 : 13.6	0 : 40 : 09.9	0 : 44 : 05.9
	sys	0 : 13 : 28.6	0 : 00 : 20.9	0 : 00 : 24.7
Textured Image 2	real	3 : 38 : 13.4	0 : 30 : 36.7	0 : 25 : 49.9
(Figure 3a)	user	1 : 47 : 42.4	0 : 11 : 48.2	0 : 25 : 13.5
	sys	0 : 12 : 55.0	0 : 00 : 35.4	0 : 00 : 15.1
Magellan Data	real	1 : 01 : 20.0	0 : 01 : 57.0	0 : 06 : 05.9
(Figure 4a)	user	0 : 12 : 57.0	0 : 01 : 31.5	0 : 03 : 29.5
	sys	0 : 03 : 27.9	0 : 00 : 03.8	0 : 00 : 09.4
IRS Data	real	2 : 36 : 48.9	—	0 : 33 : 54.8
Figure 5a)	user	0 : 45 : 42.9	—	0 : 13 : 54.4
	sys	0 : 07 : 00.5	—	0 : 00 : 13.3

The results include the time required when MLP alone is used as well as when SOM is interfaced with MLP.

* Type of time stamps given in UNIX system.

proposed system depends on the dimensionality of the SOM output layer. In some cases, the best reduced mapping requires more than two dimensions; violating this may give degraded performance. Also, if the dimensionality of SOM output lattice is increased, the overall classification time will be more. The clustering accuracy and hence the overall classification accuracy is also dependent on the width σ of the Gaussian function.

REFERENCES

Ari Visa. (1990). A texture classifier based on neural network principles. *Proceedings of the International Joint Conference on Neural Networks*, San Diego, **1**, 491–496.

Bajcsy, R. (1973). Computer description of textured surfaces. *Proceedings of the International Joint Conference on Artificial Intelligence*, Stanford, CA, August 20–23.

Berger, D. H. (1970). Texture as a discriminant of crops on radar imagery. *IEEE Transactions on Geoscience and Electronics*, **GE-8**(4), 344–348.

Bovik, A. C., Clark, M., & Geisler, W. S. (1987). Computational texture analysis using localized spatial filtering. *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, Miami Beach, FL.

Bovik, A. C., Clark, M., & Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(1), 55–73.

Campbell, F. W., & Robson, J. G. (1968). Application of Fourier analysis of the visibility of gratings. *Journal of Physiology (London)*, **197**, 551–556.

Chellappa, R., Manjunath, B. S., & Simchony, T. (1992). Texture segmentation with neural networks. In B. Kosko (Ed.), *Neural networks for signal processing*. Englewood Cliffs, NJ: Prentice-Hall.

Coggins, J. M., & Jain, A. K. (1985). A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, **3**, 195–203.

Coleman, G. B., & Andrews, H. C. (1979). Image segmentation by clustering. *Proceedings of the IEEE*, **67**(5), 773–785.

Daugman, J. G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, **20**, 847–856.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, **2**(7), 1160–1169.

Daugman, J. G. (1988). Complete discrete 2-D Gabor transforms by neural network for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36**(7), 1169–1179.

DeCosta, F. A., & Chouikha, M. F. (1992). Neural network recognition of textured images using third order cumulants as functional links. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, San Francisco, **4**, 77–80.

De Valois, R. L., Albrecht, D. G., & Thorell, L. G. (1982). Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, **22**, 545–559.

Dupaguntla, N. R., & Vemuri, V. (1989). A neural network architecture for texture segmentation and labeling. *Proceedings of the International Joint Conference on Neural Networks*, Washington DC, **1**, 127–133.

Gabor, D. (1946). Theory of communication. *Journal of the IEE*, **93**, 429–459.

Greenspan, H., Goodman, R., & Chellappa, R. (1989). Texture analysis via unsupervised and supervised learning. *Proceedings of the International Joint Conference on Neural Networks*, Seattle, **1**, 639–644.

Haralick, R. M. (1979). Statistical and structural approach to texture. *Proceedings of the IEEE*, **67**(5), 786–804.

Haralick, R. M., Shanmugam, K. S., & Distein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-3**, 610–621.

Huang, Z., & Kuh, A. (1992). A combined self-organizing feature map and multilayer perceptron for isolated word recognition. *IEEE Transactions on Signal Processing*, **40**(11), 2651–2657.

Jacobson, L. D., & Wechsler, H. (1988). Joint spatial/spatial-frequency representations. *Signal Processing*, **14**, 37–68.

Jain, A. K., & Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, **24**, 1167–1186.

Keller, J. M., Chen, S., & Crownover, R. M. (1989). Texture description and segmentation through fractal geometry. *Computer Vision, Graphics and Image Processing*, **45**, 150–166.

Knudsen, E. I., du Lac, S., & Esterly, S. D. (1987). Computational maps in the brain. *Annual Review of Neuroscience*, **10**, 41–65.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, **78**(9), 1464–1480.

Laws, K. I. (1980). *Textured image segmentation* (USCIPI Rep. 940). Image Processing Institute, University of Southern California, LA.

Leung, M., & Peterson, A. M. (1991). Multiple channel neural network model for texture classification and segmentation. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, **4**, 2677–2680.

Mallat, S. G. (1989a). A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(7), 674–693.

Mallat, S. G. (1989b). Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**(12), 2091–2110.

Marceljia, S. (1980). Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America A*, **70**(11), 1297–1300.

Pollen, D. A., & Ronner, S. F. (1983). Visual cortical neurons as localized spatial frequency filters. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-13**(5), 907–916.

Raghu, P. P., Chouhan, H. M., & Yegnanarayana, B. (1993). Multispectral texture classification using neural network. *Proceedings of the National Conference on Neural Networks*, Anna University, Madras, India, *NCNN*, 1–10.

Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural computation and self-organizing maps*. New York: Addison-Wesley Publishing Company.

Rosenfeld, A. (Ed.), (1982). *Multiresolution image processing and analysis*. New York: Springer-Verlag.

Rumelhart, D. E., Hinton, G. E., & William, R. J. (1986). Learning internal representations by error propagations. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel distributed processing Vol. I*. Cambridge, MA: MIT Press.

Sklansky, J. (1978). Image segmentation and feature extraction. *IEEE Transactions on Systems, Man and Cybernetics*, **8**, 237–247.

Turner, M. R. (1986). Texture discrimination by Gabor filters. *Biological Cybernetics*, **55**, 71–82.

Van Essen, D. (1979). Hierarchical organization and functional streams in the visual cortex. *Annual Review of Neuroscience*, **2**, 227–263.

Weszka, J. S., Dyer, C. R., & Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-6**, 269–285.

NOMENCLATURE

A	scaling factor for Gabor filter
$f(x,y)$	Gabor filtered image at pixel position (x,y)
$g(x, y, k_x, k_y, \sigma_x, \sigma_y)$	2-D Gabor filter at position (x, y), with sinusoid frequencies k_x and k_y, and Gaussian widths σ_x and σ_y
$I(x, y)$	image intensity at pixel position (x, y)
k_x	sinusoid frequency of Gabor filter in x-direction
k_y	sinusoid frequency of Gabor filter in y-direction
MLP	multilayer perceptron trained with back-propagation algorithm
M	dimensionality of SOM output node lattice
N	dimensionality of SOM input feature space, equal to the number of Gabor filters
j	the winner node in the SOM output node lattice for the input x_i
SOM	self-organizing feature map
W	set of connection weights in the SOM network
$\mathbf{w}_r$	the weight vector from input nodes of SOM to its output node r
$\mathbf{x}$	N-dimensional output vector from Gabor filter bank, which is the input vector to SOM
$\mathbf{y}_j$	M-dimensional secondary feature vector from SOM output, which is the coordinate of winner node j
X	space of input feature vectors x_i; $X \subset \mathrm{R}^{\mathrm{N}}$
Y	space of secondary feature vectors y_j from SOM; $Y \subset \mathrm{R}^{\mathrm{M}}$
σ_x	width of Gaussian of Gabor filter in x-direction
σ_y	width of Gaussian of Gabor filter in y-direction
$\sigma(t)$	a time-decreasing function used as the width of Gaussian used for the lateral interaction in SOM output layer
Φ_w	mapping function from SOM input space to its output space
$\Lambda_{y_j, y_r}(t)$	a time-decreasing function used for lateral interaction between winner node j and any other node r in SOM output layer