Appl. Math. and Comp. Sci., 1994, vol.4, No.3, 353–370

TEXTURE SEGMENTATION AND CLASSIFICATION USING NEURAL NETWORK TECHNOLOGY

MARIJKE F. AUGUSTEIJN*

Texture is an important characteristic used in the identification of objects and regions in images. Textured regions can be classified as belonging to one of a finite set of categories. This requires the supervised training of a classifier to associate the values of a set of texture features with the appropriate texture categories. Alternatively, an image can be segmented into regions showing the same texture without categorization. An unsupervised clustering technique is often used to group the feature vectors for this purpose. Neural networks can be used to achieve both the supervised texture classification as well as the unsupervised segmentation. The paper discusses how the cascade-correlation architecture can be used for both texture classification and segmentation. This supervised neural network architecture is shown to be capable to segment images containing textures not used for training the classifier. The usefulness of several texture representations is also investigated.

1. Introduction

Texture is one of the most basic characteristics of a visible surface. It has been demonstrated that texture plays an important role in human visual perception and provides important information for object recognition and scene interpretation. There are two basic approaches to texture analysis: structural and statistical. The structural approach describes textures as repeating patterns of substructures and uses rules or grammars to generate and describe these relationships. This is considered most appropriate when describing textures from high resolution images that display much regularity (Ballard and Brown, 1982). Statistical methods describe textures in terms of the spatial distribution of the gray levels in an image. This approach seems more suitable for natural textures and low resolution textures such as those seen in aerial images which have barely discernible primitives. It involves the extraction of textural features and the classification and segmentation of textured images based on these features. This paper will only be concerned with the statistical approach.

Image segmentation is the technique which partitions an image into units that are homogeneous with respect to one or more characteristics. It is often viewed as the bridge between low level image processing which involves edge detection and linking and high level object recognition and region labelling. Most image segmentation techniques use pixel gray level values as the main criterion for grouping. This is not

* Computer Science Department, University of Colorado at Colorado Springs, Colorado Springs, CO 80933–7150, USA

possible for textured surfaces which may show large variability in small-scale gray level values. Texture segmentation and classification both require feature extraction.

Many features have been proposed in the literature that measure texture properties. Textures can be described in terms of features like homogeneity, contrast, correlation and entropy. These measures can be obtained from statistical distributions of the gray level values. For example, Weszka *et al.* (1976) calculate the frequency of pixel value differences in an image fragment and base a set of texture measures on this one-dimensional distribution. Haralick *et al.* (1973) have introduced the co-occurrence matrix as a two-dimensional distribution of specific gray level occurrences. Their co-occurrence measures, calculated from these matrices, are among the most widely used textural features. Experiments, performed in the areas of psychology and biology, suggest that the probabilities of spatial gray level distributions play a role in human texture identification (Conners and Harlow, 1981). Other feature sets emphasize the frequency and direction of pattern repetition. Features of this type can be obtained from the Fourier transform of an image fragment. Weszka *et al.* (1976) have experimented with a certain kind of Fourier features and found this representation inferior to co-occurrence measures. Augusteijn *et al.* (1993), on the other hand, showed that a different variety of Fourier measures could outperform co-occurrence features. However, the Fourier transform is a global transformation of an image segment and may not be the most appropriate method to represent local gray level variability. Gabor filters measure the spectrum locally and have recently been used profusely for texture classification and segmentation. Lu *et al.* (1991), Jain and Farrokhnia (1991), and Greenspan and Goodman (1993) have all based their texture identification methods on Gabor filters. It is also possible to emphasize the stochastic nature of textures and use Markov Random Field features or fractal features as a texture characterization. Markov Random Field features were used by Khotanzad and Kashyap (1987) for texture classification and Pentland (1986) has used a fractal representation for this purpose. Alternatively, local gray level variability could be used directly as a texture representation. He and Wang (1992) have introduced the texture spectrum and have shown how this representation can be used for texture classification and segmentation. This method defines the notion of a texture unit which specifies the location of pixels in a 3×3 neighborhood with larger and smaller gray levels than the central pixel. The texture spectrum is a histogram of texture units. Augusteijn *et al.* (1993) have used a gray level averaging scheme which represents a texture by means of the gray value of a central pixel and a set of average gray values of increasingly larger neighborhoods. This representation is particularly simple to calculate.

Independent of the texture representation used, some further processing is necessary for classification and segmentation. Statistical clustering techniques are commonly employed, neural networks can be used as an alternative. The appeal of neural networks for pattern classification is based upon several considerations. They appear to perform as well as or better than other classification techniques and require no assumptions about the nature of the distribution of the pattern data. A comparison of neural networks to other methods like K-nearest neighbor and discriminant analysis has shown that neural networks can achieve equal performance using a much smaller set of training data (Hepner, 1990). Neural networks can be divided into two categories with respect to basic learning strategies. One category uses supervised learning. These networks employ a set of training patterns to establish an association between feature patterns and their corresponding categories. This association is manifested by the internal structure of the network consisting of units and weighted connections. The well-known backpropagation architecture (Rumelhart *et al.*, 1986) is the prototype of this category. A backpropagation network is a static structure requiring the researcher to specify the number of hidden layers and number of units in each of these layers. It has been found that network performance is strongly dependent on an appropriate design of this internal configuration (Baum and Haussler, 1989), but determining the correct number of internal layers and units is more an art than a science. The cascade-correlation architecture (Fahlman and Lebiere, 1990) solves this problem by allowing a network to build its own internal arrangement during training. This dynamic architecture will build a near-minimal structure sufficient for a given pattern recognition task. The other neural network category employs unsupervised learning and does not require the existence of a training set. Networks of this category cluster their input data based on pattern similarity. The Kohonen architecture (Kohonen, 1988) serves as the prototype. This network, also called the Kohonen self-organizing map, has no internal structure. It simply consists of an input layer which accepts the feature patterns and an output layer representing the clusters. It creates a topology preserving map in which units which are geometrically located near each other will learn to respond to similar input patterns. This architecture can help to gain insight into the underlying structure of the classification problem because it can show how the various textures are clustered. An example of this capability can be found in Augusteijn and Dimalanta (1992).

Both supervised and unsupervised network learning have been used for texture classification. A large literature is available on the use of neural networks for texture classification and segmentation. The following is a small sample of some more recent publications. Kirk and Pimmel (1992) employ a standard backpropagation network to classify synthetic image textures produced by a Markov process model. Scarberry and Zhang (1991) use this same architecture to categorize textures in biomedical images according to their dominant orientation. Shang and Brown (1992) use a combination of two cascaded backpropagation networks for texture classification. The first network in the cascade performs feature extraction by means of a principal component transformation, while the second network is responsible for the actual classification. Schramm and Spinnler (1992) compare the performance of backpropagation networks and restricted coulomb energy (RCE) architectures on some industrial texture segmentation tasks. An RCE architecture is configured dynamically during training. They found that the two architectures showed similar performance. Simula and Visa (1992) report on the capability of a Kohonen architecture to perform unsupervised texture classification and segmentation. Lu *et al.* (1991) use Gabor filters in combination with a Kohonen network for texture segmentation. A similar combination was employed by Greenspan and Goodman (1993). Their work is an example of a combined neural network and rule-based approach for unsupervised and supervised learning while providing probability estimates for the output classes. Hybrid networks have also been studied. Hernandez *et al.* (1992) use a hybrid of a backpropagation network and Kohonen's learning vector quantization (LVQ) method for satellite image data. LVQ is a supervised extension of Kohonen's self-organizing map. This study uses gray level values, not textures, but the network architecture could also be employed for texture classification.

2. The Cascade-Correlation Neural Network Architecture

Texture classification and segmentation are pattern recognition tasks. Pattern recognition is a major application area for neural network technology and many different architectures can be used. The experiments, described in this paper, apply cascade-correlation (Fahlman and Lebiere, 1990), and this architecture will be discussed in some more detail. Cascade-correlation is a feed-forward neural network designed to improve the slow learning characteristics of standard backpropagation. Its main distinguishing feature is its dynamic character: the network builds its internal structure incrementally, during training. Thus, the programmer need not be concerned with the appropriate number of units in the hidden layer(s) because the network itself will allocate the number of nodes required to solve the problem. The essence of the training algorithm is as follows. The initial network consists of only two layers: an input and an output layer which are completely connected. These connections are trained until no significant changes occur anymore. If, at that point, the total error is still unacceptably high, a hidden node will be allocated to further reduce the total error.

The training of hidden nodes occurs off-line. A pool of candidate hidden units is allocated. Each candidate receives activations from all input units and all hidden nodes previously inserted into the network. At first, only the input connections of the candidate units are trained. The algorithm attempts to maximize the correlation between a candidate's output and the remaining output error of the network calculated over all training patterns. The candidate whose output correlates best with this error will be able to reduce it to the greatest possible extent. This candidate is selected as the hidden node to be inserted into the network. It is placed in a separate layer above all previously allocated hidden nodes. In this manner, a cascade of hidden nodes is built. Figure 1 shows this architecture. The newly allocated hidden unit is inserted together with its trained input connections. The weights of these connections will remain fixed during the remainder of the training session. Each hidden node serves as a permanent feature detector. Once inserted, its functionality will never be corrupted by the allocation of other hidden units. This also enables incremental learning by this architecture. The training set can be partitioned into lessons. Each lesson may represent a certain aspect of the problem which can be learned by itself.

After its insertion into the network, a hidden node is connected with the output units, and these connections are now trained. All output connections are re-trained after each insertion. The actual training procedure is the quickprop algorithm, a second-order improvement to back-propagation, also developed by Fahlman (1988). The cycle of hidden node allocation and training of the output connections is repeated until the total error falls below a preset limit (successful termination) or until a preset maximum number of hidden units is reached (unsuccessful termination). There are basically two ways to measure the network's error. One measure considers the

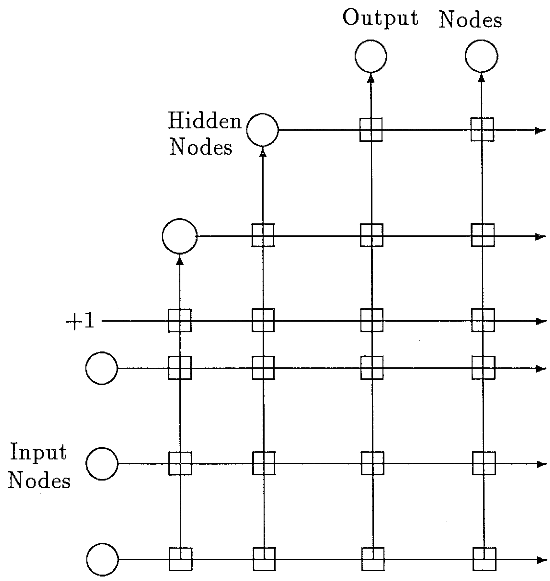


Fig. 1. The cascade-correlation architecture.

individual outputs of the network and continues training until all of them are within a preset distance from their targets. In this manner, all patterns in the training set are learned within this tolerance. The other measure considers the total distance between the network's outputs and the corresponding targets. Training is completed when the average error falls below a preset limit. In this case, some training patterns may still be misclassified after successful termination. These patterns are often outliers (exemplars, not characteristic of the texture class). Overall classification performance may be improved if the network is not forced to learn to classify these outliers correctly. A network, trained in this manner, is also likely to be more noise resistant. If the patterns in the training set are noisy, the network will learn the noise as well as the classification features. By not requiring the network to learn the noisiest patterns in the training set overall performance on a test set can be significantly improved.

3. Texture Representations

A surprisingly small number of comparative studies has been performed to determine which ones of the many texture representations are best for classification and segmentation purposes. One of the older comparisons is the study by Weszka *et al.* (1976) who compared the performance of gray level difference density measures, co-occurrence measures, a set of features derived from the Fourier transform, and run-length matrix analysis on a terrain classification task. They obtained the best performance from the co-occurrence measures. Their work is experimental in nature and performed on a very small data set. Conners and Harlow (1980) compared features on generated textures and concluded that the co-occurrence features performed better than run-length matrix analysis, gray-level difference density, or measures derived from the Fourier spectrum. This study essentially agrees with the work of

Weszka *et al.* but the results are based on theoretical analysis rather than experiments. Du Buf *et al.* (1990) compared features for image segmentation and found that co-occurrence measures performed among the best. Ohanian and Dubes (1992) compared co-occurrence features, Gabor filters, Markov random field features, and fractal features on several synthetic and natural scenes. They did not find a single, universally best, feature set, but concluded that co-occurrence measures were best for the classification of natural scenes. Augusteijn *et al.* (1993) compared gray level averaging, co-occurrence features, gray-level difference density, texture-tone analysis, Gabor filters, and a variety of features derived from the Fourier transform on satellite image data. This study also did not result in a universally best set, but showed that certain Fourier features could sometimes outperform co-occurrence measures. This result was confirmed in a different study (Augusteijn and Clemens, 1994) involving a small set of textures from the Brodatz atlas (Brodatz, 1966). However, when these experiments were repeated on a larger collection of Brodatz textures, co-occurrence measures again proved to be superior. These measures require extensive calculations and a system using them may be relatively slow. Some of the alternative representations can be calculated faster. Some of the more important texture measures are discussed in the following sections.

3.1. Co-occurrence Features

Co-occurrence features were introduced by Haralick *et al.* (1973) and are defined entirely in the spatial domain of the image. The method assumes that the texture information is adequately specified by a set of gray-tone spatial-dependence matrices, called *co-occurrence matrices.* A co-occurrence matrix measures the frequency of the simultaneous occurrence of two specified gray levels at two designated relative positions in an image fragment. These matrices can be computed for various angular relationships and distances between pixel pairs. The textural features are calculated from the two-dimensional probability densities provided by the normalized co-occurrence matrices. Haralick *et al.* (1973) define fourteen different measures. Not all researchers use the entire set. A subset of four measures: homogeneity, contrast, correlation and entropy is commonly used, and the formulae of these measures are included here as examples. If $p(i,j)$ denotes the (i,j)-th entry in a normalized co-occurrence matrix, N_g is the number of distinct gray levels in a quantized image, p_x and p_y represent the marginal-probability matrices obtained by summing the rows and columns of $p(i,j)$, respectively, and μ_x, σ_x, μ_y, σ_y are the average values and standard deviations of p_x and p_y, then these measures are calculated by means of the following formulae:

$$\text{Homogeneity}: \quad \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)^2 \tag{1}$$

$$\text{Contrast}: \quad \sum_{n=0}^{N_g-1} n^2 \left[\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\right], \qquad |i-j| = n \tag{2}$$

$$\text{Correlation}: \quad \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(ij)p(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y} \tag{3}$$

$$\text{Entropy}: \quad -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log[p(i,j)] \tag{4}$$

These measures are often combined with the average value and standard deviation of the gray levels of the image fragment.

3.2. Fourier Measures

The Fourier transform of an image fragment $f(x, y)$ is defined by:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{2\pi i(ux+vy)} f(x,y)\,\mathrm{d}x\mathrm{d}y \tag{5}$$

where u and v are the spatial frequencies in the x- and y-direction, respectively. The fast Fourier transform (FFT) algorithm, when applied to an image fragment, calculates a complex-valued spectrum of the same size as the image fragment. This algorithm performs a transformation from the spatial coordinates (x, y) to the spatial frequencies (u, v) The Fourier power spectrum is defined as $|F^2| = FF^*$, where $*$ denotes the complex conjugate. This real-valued power spectrum is used to provide the feature sets. Weszka *et al.* (1976) describe a "rings and wedges" representation. The "ring" features are averages of $|F^2|$ taken over ring-shaped regions centered at the origin. They provide a measure of texture coarseness. The "wedge" features are averages of $|F^2|$ taken over wedge-shaped regions, also centered at the origin. They represent the angular distribution of values in $|F^2|$ and are sensitive to the directionality of a texture.

Alternatively, it is possible to derive statistical measures from the Fourier power spectrum. Phillips[1] has introduced the following four features. If $I(j,k) = |F(j,k)|$ is the matrix containing the amplitudes of the spectrum and N is the number of frequency components, then these features are defined as:

$$\text{Maximum Magnitude}: \quad \max\{I(j,k) : (j,k) \neq (0,0)\} \tag{6}$$

$$\text{Average Magnitude}(A_M): \quad \sum_{j,k} I(j,k)/N \tag{7}$$

$$\text{Energy of Magnitude}: \quad \left[\sum_{j,k} I(j,k)^2\right]^{1/2} \tag{8}$$

$$\text{Variance of Magnitude}: \quad \sum_{j,k}[I(j,k) - A_M]^2/N \tag{9}$$

[1] Dr. Keith Phillips, Department of Mathematics, Univ. of Colorado at Colorado Springs, private communication

These measures can be used in combination with statistical features derived from the pixel gray level values of the image fragment. The average value and standard deviation of the pixel gray levels are good examples of such features.

Another option is to use the amplitudes of a selected set of frequencies as a feature set. Frequency selection is problem dependent. In certain cases, the power spectra of the various textures may contain characteristic frequencies; i.e., frequencies that consistently appear with high energy for certain textures. These are called the "dominant" frequencies of the texture set. The amplitude values of a set of "dominant" frequencies can be used as a feature set to distinguish between the textures. This representation was found to be extremely useful by Augusteijn *et al.* (1993). However, determining which frequencies are most characteristic for a given set of textures is not straightforward and requires extensive experimentation.

3.3. Gabor Filters

The Gabor kernel has properties which make it attractive as a receptive field for image feature extraction (Lu *et al.*, 1991). A symmetrical, two-dimensional Gabor kernel (filter) is the product of a Gaussian-shaped window and a complex exponential term:

$$\Psi(x, y;\, \sigma, u, v) = \left\{\exp \frac{-(u^2+v^2)}{2\sigma^2}(x^2+y^2)\right\} \exp i(ux+vy) \tag{10}$$

where (x, y) are the variables representing position in the spatial domain, (u, v) are the spatial frequencies, and σ is the width of the Gaussian. A different width may be used in the x- and y-direction which would give the filter a directional preference. Thus, each Gabor filter is specified by three (or four in the case of different width) parameters: σ, u, and v and is a function of (x, y) The Gabor transform, $G(x, y)$, of an image segment, $I(x, y)$, is defined as the convolution of a Gabor kernel $\Psi(x, y; \sigma, u, v)$ with $I(x, y)$:

$$G(u, v) = \int\int \Psi(x, y; \sigma, u, v) I(x, y)\, \mathrm{d}x\mathrm{d}y \tag{11}$$

The resultant transform is complex. However, only the magnitude is used because the phase information does not seem to improve classification (Lu *et al.*, 1991).

4. A Texture Classification Experiment

A set of classification experiments was performed on nine weave textures taken from the Brodatz atlas (Brodatz, 1966). The objective was to compare a variety of texture representations with respect to their classification performance. A more detailed discussion of these experiments can be found in Augusteijn and Clemens (1994); only the most important results are reported here. Twenty image fragments of size 32×32 pixels were selected from each texture. These fragments were divided into a training and a test set of equal size. Three feature sets were found to provide the best classification results. They are: a set of co-occurrence measures, a set of amplitudes of "dominant" frequencies in the Fourier transforms of the texture fragments, and a set of Gabor filters. These feature sets were used as inputs to the cascade-correlation neural network which served as the classifier. The network was configured with nine output units corresponding to the nine texture classes. The output unit showing the highest activation determined the classification of the input pattern. Each experiment consisted of five trials; i.e. a network was always trained five times on the same set of data. This is because the network weights are initialized to small random values. Thus, each trial has a slightly different starting point and the resulting trained network may show different performance. Both the range of performances and the average value are reported in percentages for each experiment.

Four co-occurrence matrices, each computing the frequency of gray-level co-occurrence at neighboring positions in four different directions (horizontal, vertical, and along the two diagonal directions of the image) were calculated for each image fragment. The set of co-occurrence features consisted of thirteen of the fourteen measures defined by Haralick *et al.* (1973). Only the maximal correlation coefficient, which requires the calculation of the second-largest eigenvalue of a co-occurrence matrix and is sometimes found unstable, was not implemented. Features were calculated from each one of a set of four matrices and then averaged over the four directions. The thirteen co-occurrence measures were combined with the average value and standard deviation of the gray level values in the image fragment. The performance of this feature set is shown in the leftmost columns of Table 1.

Many experiments were performed with Fourier measures. The weave textures show a great deal of similarity which was reflected by a resemblance in their Fourier transforms. Therefore, it was not possible to find frequencies in the Fourier spectra that were characteristic for specific textures in the set in the sense that they appeared with high amplitude values only for these textures. However, a set of frequencies could be identified that appeared with relatively high amplitudes on the average, as measured over the entire texture set. When the amplitude values of these frequencies were selected as a feature set their performance achieved better results than any other feature set used in the experiments. The results obtained from a set of fourteen of these frequencies are shown in the middle columns of Table 1.

The selection of a set of Gabor filters requires the identification of appropriate spatial frequencies and Gaussian widths. Many experiments were performed in this category. The best results were obtained from a set of dyadic modulation frequencies, similar to the ones defined in Lu *et al.* (1991). Given a set of spatial frequencies (u, v), the modulation frequency is defined as $\omega_0 = (u^2+v^2)^{1/2}$ which, when combined with the orientation $\theta = \tan^{-1}(v/u)$ and a Gaussian width σ, can be used in equation (10) to calculate a specific Gabor filter. The most successful experiment in this category employed 24 filters covering all combinations of parameter values $\omega_0 = \pi/4,\ \pi/8,\ \pi/16\ \pi/32$ and $\theta = 0, 30, 60, 90, 120$ and 150 degrees. Each filter had the same size as the image fragments and used a Gaussian width $\sigma = 10$ pixels. Features were calculated as dot products between the filters and the image fragments. The results are shown in the rightmost column of Table 1.

Table 1 shows that the set of amplitude values of selected frequencies of the Fourier transform gave the best classification performance. It is also noticeable that this set showed a very narrow performance range compared to the other two sets. Therefore, these are the preferred measures to classify the nine Brodatz weaves. The

Tab. 1. Performance in % of various texture measures on the classification of nine weaves.

Performance Range of Co-occurrence Measures	Average Performance of Co-occurrence Measures	Performance Range of Amplitudes of Dominant Frequencies	Average Performance of Amplitudes of Dominant Frequencies	Performance Range of Dyadic Gabor Filters	Average Performance of Dyadic Gabor Filters
82.2 – 91.1	86.2	93.3 – 94.4	93.8	77.8 – 85.6	82.0

main disadvantage of this method is that finding an appropriate frequency selection is very much a matter of "trial and error", requiring an appreciable amount of experimentation. Figure 2 shows the set of weave textures and the classification result obtained by means of the Fourier features and co-occurrence measures. This figure clearly shows that some weaves can be classified with great accuracy while others show relatively poor results. It is also seen that for some textures one classification method is much better than the other. The weave pattern shown in the middle of the bottom row can be classified accurately by means of the Fourier features but not by means of the co-occurrence measures. The weave pattern in the right bottom corner, on the other hand, is classified somewhat more accurately by the co-occurrence measures. The co-occurrence measures also suffer more from an edge effect than the Fourier measures as shown by the sequences of misclassifications of image fragments located near the edge of a segment. These examples clearly show the problem with identifying a universal best set of texture measures since performance depends on the collection of textures employed.

5. A Texture Segmentation Experiment

The goal of texture segmentation is to find borders between differently textured regions in an image. In order to accomplish this segmentation, it should not be necessary to have any knowledge about the actual textures present. Unfortunately, most clustering methods generally used for this purpose require an estimate of the number of different textures. The neural network approach presented here does not require any *a priori* information about the application once a network is properly trained. The cascade-correlation architecture is again used, although other neural network architectures may also be appropriate. The network is trained to compare two feature vectors originating from textured image fragments and decide if these feature vectors were generated by the same texture or by two different textures. The main objective of this research is to show that a network can be trained on a given set of textures and generalize its training to the segmentation of other textures. In this way, a general segmentation network can be obtained which can be used on any image showing textured surfaces.

The initial experiments employed the same nine weave textures used for classification. These experiments were unsuccessful. It was discovered that the network required a large texture set and many training examples before it could generalize the

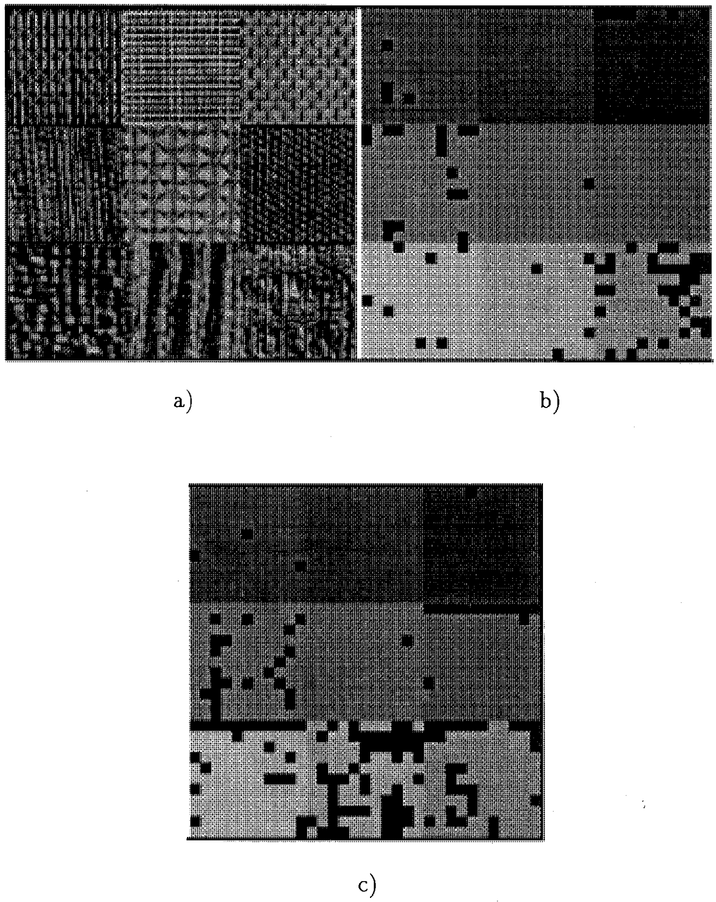


Fig. 2. Classification performance of two sets of texture measures, misclassified segments are shown in black: (a) the textures used in the experiment, (b) performance of Fourier features showing 93.4 % correct classification, (c) performance of co-occurrence measures showing 87.1 % correct classification

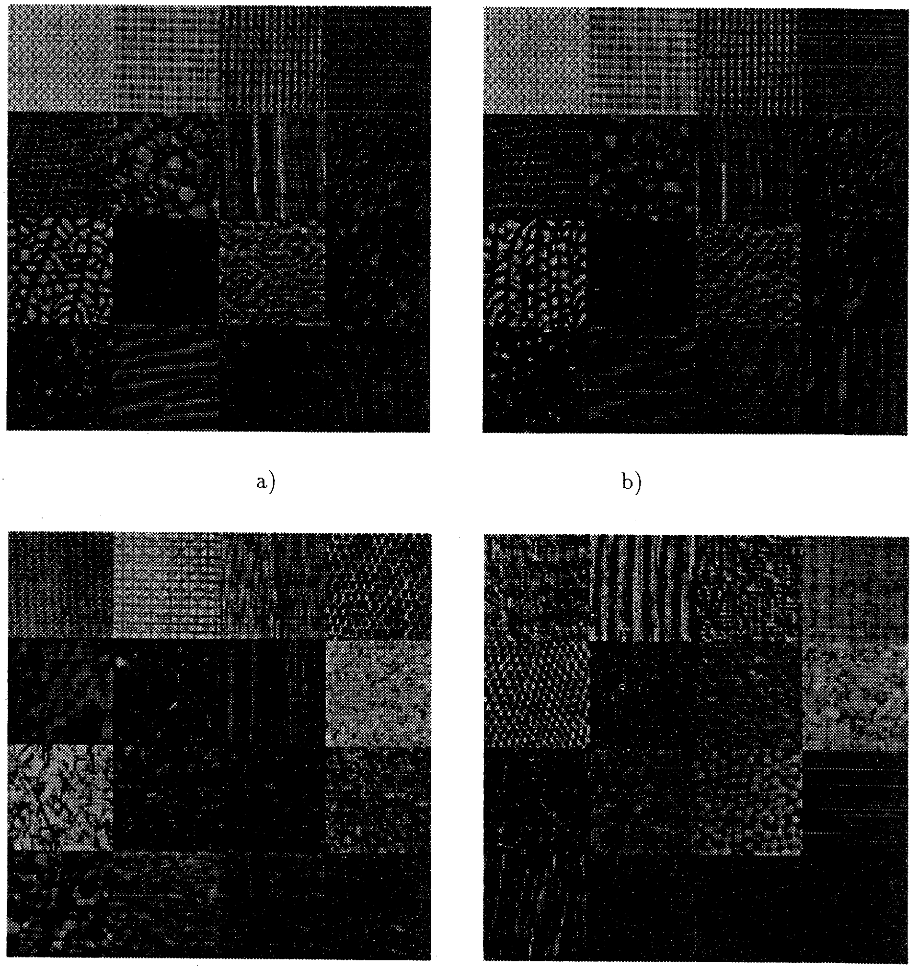
essence of the segmentation problem. The experiments that were successful used 48 different textures, provided by the Brodatz atlas (Brodatz, 1966). These textures were partitioned into three groups: one group to train the network, a second group to serve as a validation set, and a third group to test the network on new textures.

A validation set is used to prevent overtraining. Overtraining occurs when the data are noisy and the network learns the noise in the training set as well as the features that distinguish the classes. There are several ways to prevent this problem. One of the simpler methods is early termination. This approach attempts to terminate a training session before the acquisition of noise begins to dominate the learning process. This can be achieved by means of a validation set. During training, the overall error is measured over the validation set after each allocation of a hidden unit. This error is expected to pass through a minimum value when the acquisition of noise becomes greater than the acquisition of useful features. Network training is terminated at that time.

The three texture sets are shown in Fig. 3. They are referred to as set 1 (represented by two segments of each texture, see Figs. 3a and 3b), set 2 (Fig. 3c) and set 3 (Fig. 3d). Two different kinds of tests were performed. Both tests used the first texture collection of set 1 to train the networks while the textures of set 2 formed a validation set. One test measured the ability of trained networks to segment images consisting of the same textures as used in the training set. The second collection of set 1 was used for this purpose. The other test measured the ability of these same networks to generalize their segmentation capability to images consisting of textures not used to train the network. This test used the textures of set 3. All texture segments were of size 160×160 pixels patitioned into fragments of size 32×32 pixels. Texture features were extracted from each of these fragments.

Experiments were performed with several feature selections, but co-occurrence measures gave by far the best performance. The same set of four co-occurrence matrices as used for the classification experiments were calculated for each image fragment. The best results were obtained with the four measures stated in equations (1) through (4), but these measures were not averaged over the four directions in this case. Instead, the values resulting from the different matrices were concatenated, leading to a 16-component directional representation of the textures. Each set was combined with the average gray level value of the image fragment. Thus, each textured fragment was converted to a 17-dimensional feature vector.

Network training required input data obtained from two image fragments. Three kinds of input patterns were generated. The first kind simply concatenated two feature vectors, the second kind used the difference of these vectors and the third kind only employed the magnitude of the difference vector. Thus, the cascade-correlation architecture was configured with 34, 17, or 1 input unit for the three kinds of input patterns, respectively, and a single output unit. The output was trained to generate a value $+0.5$ if the network input originated from two image fragments of the same texture and a value -0.5 otherwise. It was found that the network required a large training set to learn the segmentation task reasonably well. Only 400 image fragments were available for training. But the pair combinations need not be limited to neighboring fragments. If all combinations are used a training set of 159,600 patterns will result. However, this set has a pronounced lopsided representation of the two classes. The "equal texture class" is represented by only 9,600 patterns which amounts to approximately 6% of the training data. A network trained in this manner is expected to learn the "unequal class" extremely well but will make many errors in the "equal class". To remedy this problem, it was decided to use the full set of patterns belonging to the "equal class", but only a fraction of the patterns of the "unequal class". Five image fragments were randomly selected from each of the 16 textures of set 1, and these 80 patterns were combined to generate 6000 pair patterns belonging to the "unequal class". Thus, the complete training set consisted of 15,600 patterns with a somewhat stronger representation of the "equal class".



a) b)

c) d)

Fig. 3. The texture sets used in the experiment. The top row shows the textures of set 1 used for training (a) and for testing (b). The bottom row shows set 2 (c) and set 3 (d).

It was found that many of the networks trained with concatenated vectors were not successful in reaching a low minimum error on the validation set. Five of the better performing networks in this category were selected for testing. The networks trained with difference vectors showed much more consistent performance, and the networks trained with the magnitude of the distance all performed approximately the same. The networks were tested with the patterns derived from the textures of sets 1 and 3. Any negative network output was interpreted as an indication of the existence of an edge between the two image fragments while a positive output indicated the absence of an edge. Table 2 shows network performance as percentages of correct classification. This table also includes the results of the validation set (the textures of set 2). It should be taken into account that networks were selected based on their performance on the textures of set 2. True test performance should not be measured on a validation set.

Tab. 2. Network performances in % using a set of directional co-occurrence features.

Network Input	Performance on set 1		Performance on set 2		Performance on set 3	
	Range	Average	Range	Average	Range	Average
Concatenated Vectors	87.7 – 92.2	91.2	87.4 – 89.8	88.3	82.4 – 85.4	83.8
Difference Vectors	92.0 – 93.4	92.6	88.7 – 89.4	89.0	87.7 – 89.7	88.8
Magnitude of Difference	86.9 – 87.1	87.0	89.0 – 89.3	89.2	85.8 – 86.6	86.1

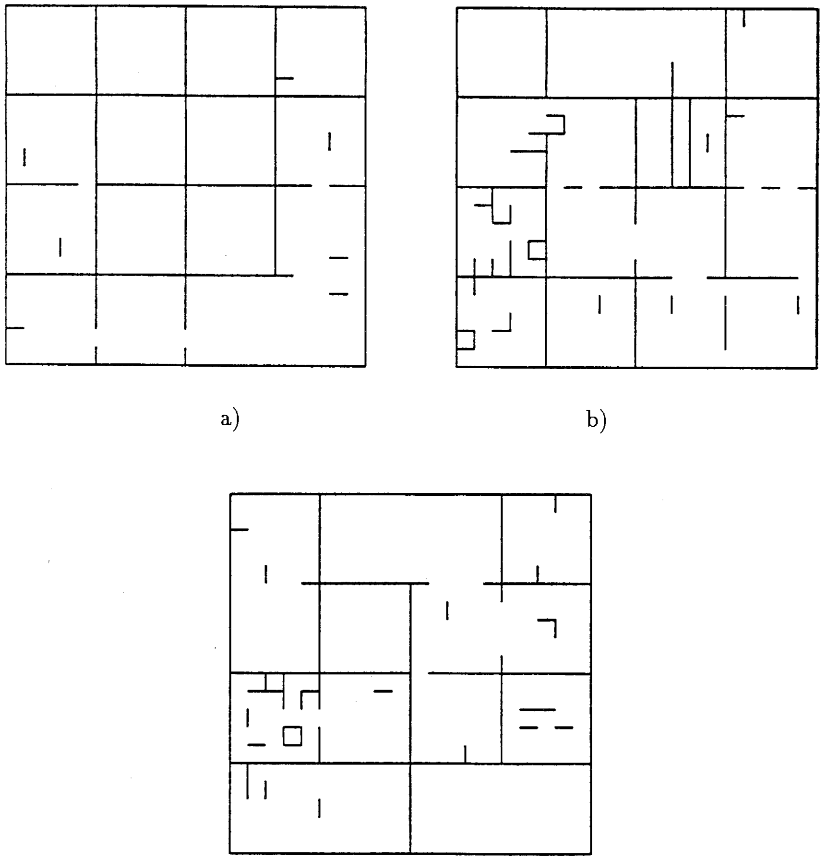
The same trained networks were also used to scan the images shown in Figs. 3b, 3c and 3d. During scanning, only feature vectors from two adjacent fragments were used to generate the input data. Some representative scanning results are shown as "edge maps" in Fig. 4. The scanning results are better than the performances shown in Table 2 because neighboring fragments of the same texture show greater similarity than fragments at a larger distance used in the test set.

6. Conclusion

Neural networks have been successful in the classification of texture sets and the segmentation of textured surfaces. A variety of architectures, employing both supervised and unsupervised training, can be used. The experiments in this paper show how the cascade-correlation architecture can be employed for both texture classification and segmentation. This architecture dynamically allocates its own hidden structure during supervised training and shows relatively fast learning compared to standard backpropagation.

Both classification and segmentation require the definition of appropriate texture measures. The same measures can be used for both tasks. A large variety of texture measures have been defined. Several comparative studies have not been conclusive in rating these measures, but many studies have indicated that co-occurrence measures (Haralick *et al.*, 1973) are among the best. The calculation of these measures tends to be slow depending on the implementation details. This paper discusses some features based on the Fourier spectrum which provide an alternative to co-occurrence measures and can often be calculated faster. However, performance may be dependent on the application domain and more study is required to assess their value.

The application of a supervised neural network architecture to accomplish general purpose segmentation is unusual. Segmentation of an unknown texture set is generally accomplished by means of clustering, and an estimate of the number of textures involved is often required. An additional disadvantage of clustering techniques is that they do not take geometrical distance of the image fragments into account. It may often be the case that fragments of the same texture that are located near each



a) b)

c)

Fig. 4. A representative set of edge maps obtained from the same network, trained with directional difference input vectors, of test set 1 (a), set 2 (b) and set 3 (c).

other in an image show greater similarity than fragments that are further apart. The neural network method discussed does not require an estimate of the number of textures present and processes neighboring image fragments. During training, the network essentially extracts a threshold from the training data. If the feature patterns of two image fragments differ by more than this threshold, then the patterns are assumed to be generated by different textures. But the network does not simply learn the geometrical distance between the feature vectors since Table 2 shows decreased performance when the magnitude of the difference vector is used as input. Apparently, not all components of the feature vector are equally useful in discriminating between the textures. The network learns to weight these components appropriately. The network segmentation worked extremely well on the set of textures also used for training. Its performance on other texture sets was less convincing, but the fact that it could segment those textures at all is remarkable. This segmentation can be followed by a post processing method which can add missing edges to a border and remove edges without continuation. The usual edge linking methods used in computer vision are also applicable to these texture edges which may lead to greatly improved segmentation.

Acknowledgment

This work was funded by the Colorado Advanced Software Institute (CASI). CASI is sponsored in part by the Colorado Advanced Technology Institute (CATI), an agency of the State of Colorado. CATI promotes advanced technology education and research at universities in Colorado for the purpose of economic development.

References

Augusteijn M.F. and Dimalanta A.S. (1992): *Feature extraction in satellite images using neural network technology.* — Telematics and Informatics, v.9, No.3/4, pp.131–144.

Augusteijn M.F., Clemens L.E. and Shaw K.A. (1993): *Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier.* — Technical Report, EAS–CS–93–8, Dept. of Computer Sciences, University of Colorado at Colorado Springs.

Augusteijn M.F. and Clemens L.E. (1994): *A performance evaluation of texture measures for image classification and segmentation using the cascade-correlation architecture.* — Proc. of IEEE Int. Conf. *Neural Networks*, Orlando, Florida, v.VII, pp.4300–4305.

Ballard D.H. and Brown C.M. (1982): *Computer Vision.* — Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Baum E.B. and Haussler D. (1989): *What size net gives valid generalization?* — Neural Computation, v.1, No.1, pp.151–160.

Brodatz P. (1966): *Textures: A Photographic Album for Artists and Designers.* — New York: Dover Publications, Inc.

Du Buf J.M.H., Kardan M. and Spann M. (1990): *Texture feature performance for image segmentation.* — Pattern Recognition, v.23, No.3, pp.291–309.

Conners R.W. and Harlow C.A. (1980): *A Theoretical comparison of texture algorithms.* — IEEE Trans. Pattern Analysis and Machine Intelligence, v.2, No.3, pp.204–222.

Conners R.W. and Harlow C.A. (1981): *Toward a structural textural analyzer based on statistical methods.* — In: Image Modeling (A. Rosenfeld Ed.), New York: Academic Press, pp.29–61.

Fahlman S.E. (1988): *Faster-learning variations on back-propagation: an empirical study.* — Proc. of the 1988 *Connectionist Models Summer School* (D. Touretzky, Ed.), San Mateo: Morgan Kaufmann, pp.38–51

Fahlman S.E. and Lebiere C. (1990): *The cascade-correlation learning architecture.* — In: Advances in Neural Information Processing Systems 2 (D. Touretzky, Ed.), San Mateo: Morgan Kaufmann, pp.524–532.

Greenspan H.K. and Goodman R. (1993): *Remote sensing image analysis via a texture classification neural network.* — In: Advances in Neural Information Processing Systems 5, (S.J. Hanson, J.D. Cowan and C.L. Giles Eds.), San Mateo: Morgan Kaufmann, pp.425–432.

Haralick R.M., Shanmugam K. and Dinstein I. (1973): *Textural features for image classification.* — IEEE Trans. Systems, Man and Cybernetics, v.3, No.6, pp.610–621.

He D. and Wang L. (1992): *Detecting texture edges from images.* — Pattern Recognition, v.25, No.6, pp.595–600.

Hepner G.F. (1990): *Artificial neural network classification using a minimal training set: comparison to conventional supervised classification.* — Photogrammetric Engineering and Remote Sensing, v.56, No.4, pp.469–473.

Hernandez R., Varfis A., Kanellopoulos I. and Wilkinson G. (1992): *Development of MLP/LVQ hybrid networks for classification of remotely-sensed satellite images.* — In: Artificial Neural Networks 2 (Aleksander I. and Taylor J. Eds), Elsevier Science Publishers B.V., pp.1193–1196.

Jain A.N. and Farrokhnia F. (1991): *Unsupervised texture segmentation using gabor filters.* — Pattern Recognition, v.24, No.12, pp.1167–1186.

Khotanzad A. and Kashyap R. (1987): *Feature selection for texture recognition based on image synthesis.* — IEEE Trans. Syst. Man Cybern., v.17, No.6, pp.1087–1095.

Kirk J.W. and Pimmel R.L. (1992): *Texture classification using multilayer neural networks with backpropagation training.* — In: Intelligent Engineering Systems through Artificial Neural Networks v.2 (Dagli C.H., Burke L.I. and Shin Y.C., Eds), New York: ASME Press, pp.529–534.

Kohonen T. (1988): *The neural phonetic typewriter.* — Computer, v.21, No.1, pp.11–22.

Lu S., Hernandez J.E. and Clark G.A. (1991): *Texture segmentation by clustering of gabor feature vectors.* — Int. J. Conf. Neural Networks, Seattle, WA, USA, v.I, pp.683–687.

Ohanian P.P. and Dubes R.C. (1992): *Performance evaluation for four classes of textural features.* — Pattern Recognition, v.25, No.8, pp.819–833.

Pentland A.P. (1986): *Shading into texture.* — Artif. Intell. v.29, No.2, pp.147–170.

Rumelhart D.E., Hinton G.E. and Williams R.J. (1986): *Learning internal representations by error propagation.* — In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, v.1: Foundations (D.E. Rumelhart and J.L. McClelland, Eds), Cambridge, Massachusetts: MIT Press, pp.318–362.

Scarberry R.E. and Zhang Z. (1991): *Determining texture orientation in biomedical images using artificial neural network.* — In: Intelligent Engineering Systems through Artificial Neural Networks (Dagli C.H., Kumara S.R.T. and Shin Y.C., Eds), New York: ASME Press, pp.351–356.

Schramm U. and Spinnler K. (1992): *Neural network image segmentation for automated visual inspection.* — In: Artificial Neural Networks 2 (Aleksander I. and Taylor J., Eds), Elsevier Science Publishers B.V., pp.1509–1512.

Shang C. and Brown K. (1992): *Texture image classification based on interframe principal features.* — In: Artificial Neural Networks 2 (Aleksander I. and Taylor J., Eds), Elsevier Science Publishers B.V., pp.1173–1176.

Simula O. and Visa A. (1992): *Self-organizing feature maps in texture classification and segmentation.* — In: Artificial Neural Networks 2 (Aleksander I. and Taylor J., Eds), Elsevier Science Publishers B.V., pp.1621–1628.

Weszka J.S., Dyer C.R. and Rosenfeld A. (1976): *A comparative study of texture measures for terrain classification.* — IEEE Trans. Systems, Man, and Cybernetics, v.6, No.4, pp.269–285.