Classification of Color Textures with Random Field Models and Neural Networks

Orlando J. Hernandez, John Cook, Michael Griffin, Cynthia De Rama, and Michael McGovern
E-mail: {hernande, cook5, griffin2, derama2, mcgover4}@tcnj.edu
Department Electrical & Computer Engineering, The College of New Jersey
Ewing, New Jersey 08628-0718, USA

ABSTRACT

A number of texture classification approaches have been developed in the past but most of these studies target gray-level textures. In this work, novel results are presented on Neural Network based classification of color textures in a very large heterogeneous database. Several different Multispectral Random Field models are used to characterize the textures. The classifying features are based on the estimated parameters of these model and functions defined on them. The approach is tested on a database of 73 different color textures classes. The advantage of utilizing color information is demonstrated by converting color textures to gray-level ones and classifying them using Grey Level Co-Occurrence Matrix (GLCM) based features.

Keywords: Color Texture, Color Texture Features, Mutispectral Random Field Models, Texture Classification

1. INTRODUCTION

Classification of textures has become a significant topic of research. There are many different approaches that have been studied due to the usage of different models for extracting features from texture images. From past research, it has been observed that classification of color textures improves the accuracy of classification over gray-level. In the experiments preformed in this work, both color and gray-level models were used to determine the features of the images. The features extracted were then used with a neural network for the classification of the texture images, and new results are presented for this combination of features, classification methodology, size of database, and number of classes.

The color texture models used for feature extraction are Multispectral Simultaneous Autoregressive (MSAR), Multispectral Markov Random Field (MMRF), and the Pseudo-Markov Random Field (PMRF) model. For each of the color texture models, a half and full neighborhood set is used. The features obtained are used by a neural network to distinguish the 73 different texture classes present in a very large heterogeneous database with minimum training of the neural network, and these results are the main novel contribution of this work.

The advantage of considering color in texture classification is also demonstrated in this work. An equivalent gray-level database is created for the color database used in this study. The gray-level textures are then classified using features derived from the GLCM with distances of one and two pixels, and the classification performance of the neural network is compared to that obtained for color textures. There is considerable gain in classification accuracy indicating that color information does provide substantial advantage to the recognition task for a very large heterogeneous database with minimum training of the neural network. The overall approach is depicted in Fig. 1.

Some Recent and Related Studies

Some recent examples of color texture classification or the involvement of Neural Networks in such task are the works presented in [1], [2], [3], and [4]. These papers focus on using the pure color vectors as the feature set, while they diverge on the classification methodology employed (histogram, transform, filtering, co-occurrence, neural network), as well as the level of comparison of the results with those for grey-level imagery. Also in another study [5], specific techniques have been developed recently to exploit the large amount of spectral and spatial information present in color textures to optimize filters aimed at the execution of illumination-invariant color texture discrimination. Other recent approaches to color texture classification also include the taking of perceptual information models into account [6], as well as the usage of random environment models [7].

The texture segmentation algorithm in [8] considers features extracted with a 2-D moving average (MA) approach. The 2-D MA model represents a texture as an output of a 2-D finite impulse response (FIR) filter with simple input process. The 2-D MA model is used for modeling both isotropic and anisotropic textures. The maximum likelihood (ML) estimator of the 2-D MA model is used as texture features. Supervised and unsupervised texture segmentation are considered. The texture features extracted by the 2-D MA modeling approach from sliding windows are classified with a neural network for supervised segmentation, and are clustered by a fuzzy clustering algorithm for unsupervised texture segmentation.

Oja and Valkealahti [9] have worked with texture feature sets based on second-order co-occurrences of gray levels of pixel pairs. An extension of the co-occurrences to higher orders is prevented by the large size of the multidimensional arrays, so the higher-order co-occurrences are quantized by the Self-Organizing Map, called the Co-occurrence Map. This allows a flexible two-dimensional representation of co-occurrence histogram of any order. Experiments with natural grey level and color textures showed that the method is effective in texture classification and segmentation.

2. IMAGE DATABASE CONSTRUCTION

For our study, the VisTex database was used [10]. These are prepackaged images and considered as one of the standard image databases for color texture processing. It is comprised of many reference textures and real-world textures. A real-world texture is an image where various textures appear. A reference natural texture is basically homogeneous, and this is what this study employs. This distinction is made in Fig. 2.

From the VisTex database, a maximum of 73 texture classes were able to be extracted. Database images with a size of 512 x 512 pixels were decimated into 64 x 64 pixels sub-images to populate the final database. Therefore, each larger texture image was fragmented into 64 square sub-images as shown in Fig. 3.

So each 512 x 512 image supplies 64 texture samples. There are a total number of 120 images that can be used from the VisTex database, which yielded a database of 7,680 texture images with 73 texture classes. With this approach, not all texture classes have the same number of texture image members, but the average number of samples or members per texture class is 105.

After the color database has been generated, there must be a grey scale counterpart to it for GLCM. A second database of gray-level textures is generated from this color database by converting all of the 7,680 color images into gray-level ones. This process is carried out using the following linear RGB to CIE luminance conversion [11].

$$Y_{CIE} = \lfloor 0.2125\,R + 0.7154\,G + 0.0721\,B + 0.5 \rfloor \qquad (1)$$

The database is then separated into a "training" set and a "test" set. The training set has about a randomly chosen 1/3 of all the images available, while the test set has the rest of the images in the database as shown in Fig. 4. The training set is used to train the neural network, and once the neural network's weights have been set, results are obtained from the test set.

3. MULTISPECTRAL RANDOM FIELD MODELS

Multispectral Random Field Models are the generalization of the gray-level random field models. They were initially developed in [12], [13]. These models are capable of characterizing color textures and are able to synthesize a color texture from the estimated parameters of the model fitted to it [13], [12]. In this work, we utilize three such models for the classification task.

Multispectral Simultaneous Autoregressive (MSAR) Model

The Multispectral Simultaneous Autoregressive (MSAR) model is the first considered model. A pixel location within a two-dimensional M x M lattice is denoted by **s** = (i, j), with i, j being integers from the set J = {0, 1, …, M-1}. The set of all lattice locations is defined as **Ω** = {**s** = (i, j) : i, j ∈ J}. The MSAR model relates each lattice position to its neighboring pixels, both within and between image planes, according to the following model equation:

$$y_i(\mathbf{s}) = \sum_{j=1}^{P} \sum_{\mathbf{r} \in N_{ij}} \theta_{ij}(\mathbf{r})\, y_i(\mathbf{s} \oplus \mathbf{r}) + \sqrt{\rho_i}\; w_i(\mathbf{s}), \quad i = 1 \ldots P \qquad (2)$$

with,

$y_i(\mathbf{s})$ = Pixel value at location s of the ith plane

s and **r** = two dimensional lattices

P = number of image planes (for color images, P = 3, representing: Red, Green, and Blue planes)

N_{ij} = neighbor set relating pixels in plane i to neighbors in plane j (only interplane neighbor sets, i.e. $N_{ij,\, i \neq j}$, may include the (0,0) neighbor)

θ_{ij} = coefficients which define the dependence of $y_i(s)$ on the pixels in its neighbor set N_{ij}

ρ_i = noise variance of image plane i

$w_i(\mathbf{s})$ = i.i.d. random variables with zero mean and unit variance

⊕ denotes modulo M addition in each index (a toroidal lattice structure is assumed so a complete neighbor set could be defined for pixels on the boundary of the image)

The image observations are assumed to have zero mean (i.e., the sample mean is computed and subtracted from all pixels).

The parameters associated with the MSAR model are **θ** and **ρ** vectors which collectively characterize the spatial interaction between neighboring pixels within and between color planes. A least-squares (LS) estimate of the MSAR model parameters is obtained by equating the observed pixel values of an image to the expected value of the model equations. This leads to the following estimates [13], [12]:

$$\hat{\theta}_i = \left[\sum_{\mathbf{s} \in \mathbf{\Omega}} \mathbf{q}_i(\mathbf{s})\, y_i(\mathbf{s})\right] \left[\sum_{\mathbf{s} \in \mathbf{\Omega}} \mathbf{q}_i(\mathbf{s})\, \mathbf{q}_i^T(\mathbf{s})\right]^{-1} \qquad (3)$$

and

$$\hat{\rho}_i = \frac{1}{M^2} \sum_{\mathbf{s} \in \mathbf{\Omega}} \left(y_i(\mathbf{s}) - \hat{\boldsymbol{\theta}}_i^T\, \mathbf{q}_i(\mathbf{s})\right)^2 \qquad (4)$$

where:

$$\boldsymbol{\theta}_i = \left[\boldsymbol{\theta}_{i1}^T\, \boldsymbol{\theta}_{i2}^T \cdots \boldsymbol{\theta}_{iP}^T\right]^T \qquad (5)$$

$$\mathbf{q}_i(\mathbf{s}) = \left[\mathbf{y}_{i1}^T(\mathbf{s})\, \mathbf{y}_{i2}^T(\mathbf{s}) \cdots \mathbf{y}_{iP}^T(\mathbf{s})\right]^T \qquad (6)$$

$$\mathbf{y}_{ij}(\mathbf{s}) = \mathrm{col}\left\{y_j(\mathbf{s} \oplus \mathbf{r}) : \mathbf{r} \in N_{ij}\right\} \qquad (7)$$

Multispectral Markov Random Field (MMRF) Model

The second kind of model considered is the Multispectral Markov Random Field (MMRF) Model. A multispectral image may be considered Markovian with respect to its neighbor set if it has the following property:

p($y_i(\mathbf{s})$ | all other image observations) = p($y_i(\mathbf{s})$ | neighborhood observations)

Because the conditional distribution of $y_i(\mathbf{s})$ given all other observations and $y_i(\mathbf{s})$ given the neighborhood observations are the same, the best linear estimator of the observed values may be written in terms of the neighborhood observations as:

$$y_i(\mathbf{s}) = \sum_{j=1}^{P} \sum_{\mathbf{r} \in N_{ij}} \theta_{ij}(\mathbf{r})\, y_i(\mathbf{s} \oplus \mathbf{r}) + e_i(\mathbf{s}), \quad i = 1 \ldots P \qquad (8)$$

where the estimation error

$$e_i(\mathbf{s}) = \sum_{j=1}^{P} \sum_{\mathbf{r} \in \mathbf{\Omega}} c_{ij}(\mathbf{r})\, w_j(\mathbf{s} \oplus \mathbf{r}), \quad i = 1 \ldots P \qquad (9)$$

is a stationary noise sequence with unit variance i.i.d variates, $w_i(\mathbf{s})$ for some choice of c_{ij}'s.

Since the resulting system of equations that could be employed to obtain a LS estimate of the model parameters is nonlinear, an approximate LS estimate approach is employed here. This is an iterative method that involves repeatedly solving the pair of equations [13], [12]:

$$\hat{\boldsymbol{\theta}}_{n+1} = \left[\sum_{\mathbf{s} \in \mathbf{\Omega}} \mathbf{Q}_n(\mathbf{s})\, \mathbf{Q}_n^T(\mathbf{s})\right]^{-1} \left[\sum_{\mathbf{s} \in \mathbf{\Omega}} \mathbf{Q}_n(\mathbf{s})\, \mathbf{y}(\mathbf{s})\right], \quad n = 0,1,2,\ldots \qquad (10)$$

and

$$\hat{\boldsymbol{\rho}}_{n+1} = [\hat{\rho}_1\ \hat{\rho}_2\ \hat{\rho}_3]^T = \frac{1}{M^2}\sum_{\mathbf{s}\in\boldsymbol{\Omega}}\left[\mathbf{y}(\mathbf{s}) - \mathbf{Q}_n^T(\mathbf{s})\hat{\boldsymbol{\theta}}\right]^2, \quad n = 0,1,2,\ldots \tag{11}$$

where

$$\mathbf{Q}(\mathbf{s}) = \begin{bmatrix} \mathbf{y}_{11}^T(\mathbf{s}) & \mathbf{y}_{12}^T(\mathbf{s}) & \mathbf{y}_{13}^T(\mathbf{s}) & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \frac{\rho_2}{\rho_1}\mathbf{y}_{21}^T(\mathbf{s}) & \mathbf{0}^T & \mathbf{y}_{22}^T(\mathbf{s}) & \mathbf{y}_{23}^T(\mathbf{s}) & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \frac{\rho_3}{\rho_1}\mathbf{y}_{31}^T(\mathbf{s}) & \mathbf{0}^T & \frac{\rho_3}{\rho_2}\mathbf{y}_{32}^T(\mathbf{s}) & \mathbf{y}_{33}^T(\mathbf{s}) \end{bmatrix}^T \tag{12}$$

and

$$\mathbf{y}_{ij}(\mathbf{s}) = \begin{cases} \operatorname{col}\{y_i(\mathbf{s}\oplus\mathbf{r}) + y_i(\mathbf{s}\oplus-\mathbf{r}) : \mathbf{r}\in N_{ii}\}, & i = j \\ \operatorname{col}\{y_j(\mathbf{s}\oplus\mathbf{r}) : \mathbf{r}\in N_{ij}\}, & i < j \\ \operatorname{col}\{y_j(\mathbf{s}\oplus-\mathbf{r}) : \mathbf{r}\in N_{ji}\}, & i > j \end{cases} \tag{13}$$

$\mathbf{Q}(\mathbf{s})$ is evaluated using $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}_n$ and $\hat{\boldsymbol{\rho}}_0$ is taken as [1 1 1]^T . The iteration terminates when convergence is obtained, i.e. subsequent iterations fail to produce significant changes in $\boldsymbol{\theta}$. This approach works well in practice, typically requiring less than 10 iterations to obtain the LS estimate of MMRF model parameters.

Pseudo-Markov Random Field (PMRF) Model

The third model is the Pseudo-Markov Random Field (PMRF) model which has the same model equations as the MMRF:

$$y_i(\mathbf{s}) = \sum_{j=1}^{P}\sum_{\mathbf{r}\in N_{ij}}\theta_{ij}(\mathbf{r})\, y_j(\mathbf{s}\oplus\mathbf{r}) + e_i(\mathbf{s}), \quad i = 1\ldots P \tag{14}$$

However, by restricting the correlation structure of the stationary noise sequences $\{e_i(\mathbf{s})\}$ of the PMRF model, it lends itself to a linear LS estimation scheme of its model parameters, instead of the non-linear iterative approach used for the MMRF model. The assumed correlation structure for $\{e_i(\mathbf{s})\}$ is:

$$v_{ij}(\mathbf{r}) = E\{e_i(\mathbf{s})\, e_j(\mathbf{s}\oplus\mathbf{r})\} = \begin{cases} -\theta_{ij}(\mathbf{r})\sqrt{\rho_i\,\rho_j}, & \mathbf{r}\in N_{ij} \\ \rho_j, & \mathbf{r} = 0 \text{ and } i = j \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

The LS estimate of the PMRF model parameters is given by [13], [12]:

$$\hat{\boldsymbol{\theta}} = \left[\sum_{\mathbf{s}\in\boldsymbol{\Omega}}\mathbf{Q}(\mathbf{s})\,\mathbf{y}(\mathbf{s})\right]\left[\sum_{\mathbf{s}\in\boldsymbol{\Omega}}\mathbf{Q}(\mathbf{s})\,\mathbf{Q}^T(\mathbf{s})\right]^{-1} \tag{16}$$

and

$$\hat{\boldsymbol{\rho}} = [\hat{\rho}_1\ \hat{\rho}_2\ \hat{\rho}_3]^T = \frac{1}{M^2}\sum_{\mathbf{s}\in\boldsymbol{\Omega}}\left[\mathbf{y}(\mathbf{s}) - \mathbf{Q}^T(\mathbf{s})\hat{\boldsymbol{\theta}}\right]^2 \tag{17}$$

where:

$$\mathbf{Q}(\mathbf{s}) = \begin{bmatrix} \mathbf{y}_{11}^T(\mathbf{s}) & \mathbf{y}_{12}^T(\mathbf{s}) & \mathbf{y}_{13}^T(\mathbf{s}) & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{y}_{21}^T(\mathbf{s}) & \mathbf{0}^T & \mathbf{y}_{22}^T(\mathbf{s}) & \mathbf{y}_{23}^T(\mathbf{s}) & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{y}_{31}^T(\mathbf{s}) & \mathbf{0}^T & \mathbf{y}_{32}^T(\mathbf{s}) & \mathbf{y}_{33}^T(\mathbf{s}) \end{bmatrix}^T \tag{18}$$

and

$$\mathbf{y}(\mathbf{s}) = [y_1(\mathbf{s})\quad y_2(\mathbf{s})\quad y_3(\mathbf{s})]^T = E\{\mathbf{y}(\mathbf{s})\,|\,\boldsymbol{\theta}\} = \mathbf{Q}^T(\mathbf{s})\,\boldsymbol{\theta}, \quad \mathbf{s}\in\boldsymbol{\Omega} \tag{19}$$

Neighborhood Sets

The parameter that estimates the coefficients which define the dependence of a pixel value at a location on a plane on the pixels in its neighborhood set is dependent on the type of neighborhood set used. In this experiment, two different types of neighborhood sets were used, a full and a half neighborhood, for each model. The full neighborhood set consists of the eight elements that surround the pixel being observed, and the half neighborhood consists of the four elements of the surrounding eight in the vertical and horizontal directions. These neighborhoods are symmetric, and they are shown in Fig. 5.

Color Content Characterization

In addition to modeling color texture, the general color content of the image is also important. Additional features focusing on the color alone are also considered. This is done using the sample mean of the pixel values in the red, green, and blue (RGB) planes. The defined feature vector is:

$$\mathbf{f}_C = \left\{\frac{\hat{\mu}_r}{\hat{\mu}_g}, \frac{\hat{\mu}_r}{\hat{\mu}_b}\right\} \tag{20}$$

with $\hat{\mu}_i$s being the sample mean of the respective color component. The reason for using ratio of color means instead of color means themselves is that such a ratio is illumination invariant. Assuming that the observed value at each pixel is a product of illumination and spectral reflectance, the ratios of the color means are invariant to uniform changes in illumination intensity (i.e. the power of the illumination source changes uniformly across the spectrum). This kind of uniform change would cause each $\hat{\mu}_i$ to change by the same scale factor making the defined ratios invariant to illumination changes. This property makes the color-content features more robust. In the event that the denominator of any of the ratios of the color means goes to zero, the color mean with a value of zero is changed to a value of one to avoid the mathematical exception of dividing by zero. This case; however, is very unlikely, since we are dealing with textures and natural images that do not tend to have large areas with a solid color extreme.

Features for Color Textures

The features that are used for classification of color textures are the color content as discussed previously and the estimated θ and ρ parameters of the discussed multispectral random field models. For the MSAR model with the full neighborhood set, the estimated parameters lead to a feature set with 40 features. Using the half neighborhood set, 22 features were extracted from the images. Because of the similarities of the parameters of the MMRF and PMRF models, the resulting amount of features created using the full and half neighborhoods were the same for each model respectively. With these two models, 28 features were created using a full neighborhood set, and using a half neighborhood set, 16 features were created for each image. The estimated θ parameters are used directly whereas ratios of the ρ parameters of different color planes in the form of $\frac{\rho_r}{\rho_g}$ and $\frac{\rho_r}{\rho_b}$ are utilized. The justification for using ratios is the same as the one explained to use the rations of the color means. The parameter vectors θ_{ij} are a function of the selected neighbor sets. Thus we have for the MSAR model

$$\mathbf{f}_{MSAR} = \left\{\mathbf{f}_C, \frac{\rho_r}{\rho_g}, \frac{\rho_r}{\rho_b}, \boldsymbol{\theta}_{rr}, \boldsymbol{\theta}_{rg}, \boldsymbol{\theta}_{rb}, \boldsymbol{\theta}_{gr}, \boldsymbol{\theta}_{gg}, \boldsymbol{\theta}_{gb}, \boldsymbol{\theta}_{br}, \boldsymbol{\theta}_{bg}, \boldsymbol{\theta}_{bb}\right\} \quad (21)$$

with each θ_{ij} representing a vector with a number of elements equal to the number of neighbors in the neighborhood set parameters.

For the MMRF and PMRF models, the θ_{ij} of symmetric neighbors are taken to be equal in these models. Consequently, only half of θ_{ij} parameters are used.

$$\mathbf{f}_{MMRF} = \left\{\mathbf{f}_C, \frac{\rho_r}{\rho_g}, \frac{\rho_r}{\rho_b}, \boldsymbol{\theta}_{rr}, \boldsymbol{\theta}_{rg}, \boldsymbol{\theta}_{rb}, \boldsymbol{\theta}_{gg}, \boldsymbol{\theta}_{gb}, \boldsymbol{\theta}_{bb}\right\} \quad (22)$$

and

$$\mathbf{f}_{PMRF} = \left\{\mathbf{f}_C, \frac{\rho_r}{\rho_g}, \frac{\rho_r}{\rho_b}, \boldsymbol{\theta}_{rr}, \boldsymbol{\theta}_{rg}, \boldsymbol{\theta}_{rb}, \boldsymbol{\theta}_{gg}, \boldsymbol{\theta}_{gb}, \boldsymbol{\theta}_{bb}\right\} \quad (23)$$

Table 1 summarizes the number of features obtained for each model and neighborhood set.

4. FEATURES FOR GREY-LEVEL TEXTURES

The grey-level version of each of the color textures considered in this study is also generated using the conversion method discussed previously. In this study, GLCM was used. GLCM takes an image with a known number of grey-levels and calculates the frequency of the gradient of color between pixels; effectively calculating a 2-D histogram. After this histogram is calculated, the values are divided by the total number of pixels so that the GLCM becomes a listing of relative probabilities. Finally, the probability matrix is read and manipulated such that features corresponding to texture properties of the image can be calculated.

To calculate the GLCM, the image is analyzed by starting at the upper left-hand corner, at location (0, 0), and then sequentially read in a raster scan fashion. An angle of analysis for the second pixel must also be used: 0, 45, 90, and 135 degrees were used. Because the GLCM calculation uses absolute values to determine the grey-level gradient, angles of 180, 225, 270, and 315 degrees are obtained from their symmetric counterparts. Additionally, using these angles makes the features somewhat rotation invariant. In this project, distances of 1 and 2 pixels were used, since they are typical values for the GLCM technique. After the $\mathrm{GLCM}(d, \theta)$ matrices are obtained, then the probability matrices $P(x, y, d, \theta)$ are calculated.

Finally, features are derived from the P matrix. While GLCM methods have been devised for over 50 different types of features, for this experiment, Contrast, Entropy, Energy, and Correlation were chosen. The following equations illustrate the formulae used for feature calculation:

Contrast $$\sum_{GLCM} (i-j)^2\, P(i,j,d,\theta) \quad (24)$$

Energy $$\sum_{GLCM} P(i,j,d,\theta)^2 \quad (25)$$

Entropy $$-\sum_{GLCM} P(i,j,d,\theta)\, \log_{10} P(i,j,d,\theta) \quad (26)$$

Correlation $$\frac{\sum_{GLCM} (i-\mu_x)(j-\mu_y)\, P(i,j,d,\theta)}{\sigma_x \sigma_y} \quad (27)$$

Calculating features from GLCM methods is a relatively straightforward and fast algorithm. When GLCM is employed, the image must be of a minimum size (32 by 32), or else there will not be sufficient information to derive an accurately classifiable feature set.

5. CLASSIFICATION METHOD

To achieve the classification, a neural network was selected, which was trained with the training set of the database (about 30% of the total number of images). A fully connected 4-layer network is being used in this study; it employs 2 hidden layers, and the hyperbolic tangent was used as the decision function. The number of nodes in the input and output layer are set directly by the input database. The input layer size is determined by the number of features being presented to the network. The output layer size is determined by the number of classes existing in the input set. In this study, a total of 73 classes are present in the database. The number of nodes in the hidden layers is not explicitly specified by the input data set. A larger number of nodes may increase the performance of the network but it will increase complexity and calculation time. The rule for the size of each of the hidden layers is given by the following equation [8]:

$$\mathrm{HL\ SIZE} = 2 \times F + 1 \quad (28)$$

where F is the number of features input to the network. The initial value of each weight in the network is set at random in the following symmetric range [14]:

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i}\right) \quad (29)$$

where F_i is the fan-in of the node. This allows the range for possible weights to have different values for each layer. An example of the neural network used is shown in Fig. 6.

6. CLASSIFICATION RESULTS

The neural network classification results are shown in Table 2. The number of correctly classified samples out of the total of 5,120 (~70% of the total database) tested samples is shown for the color and gray-level images for features derived from different models.

These results demonstrate that very high classification (in the 90% range) can be achieved for 64×64 color texture images using features of MSAR, MMRF, or PMRF models with a simple 8-pixels neighborhood. This high classification rates are obtained on a very large heterogeneous color texture database with minimal training of a neural network. Considering the results for both neighborhoods, it may be concluded that the features based on the full neighborhood, rather than those based on the half neighborhood, would be the best choice for the classification task.

As for classification results of the gray-level counterpart images, the accuracy rates are well below 50% for both sets of GLCM based features. By comparing the classification results of color images to their gray-level converted counterparts, the advantage of using color becomes apparent. The color results are clearly better. While the color textures are classified in the 90% range, the rate for gray-level images is below the 50% range. When only grey-level information is considered for texture classification, the textural details within a single plane can become fuzzy, and interaction between different image planes becomes more dominant. Inter-plane interactions are efficiently captured by the multispectral models causing them to perform better than the single plane, gray-level features.

7. CONCLUSIONS

In this work three different multispectral random field models, MSAR, MMRF, and PMRF, are used for supervised color texture classification. Features are defined on the estimated parameters of these models fitted to the images. These models capture both inter-plane and intra-plane interactions of image pixels resulting in richer characterization of the image compared to the gray-level only GLCM based features. The performance is tested on a large database of 5,120 images and 73 classes. It is shown that a small and compact neighbor set is all that is needed for the classification task. In a neural network based classification scheme and utilizing normalized features, very high classification in the 90% range is obtained for 64×64 images.

To show the advantage of incorporating color (multi-plane activity) in the classification task, gray-level counterparts of the color texture images are created and classified using GLCM based features. It is shown that the classification result is inferior to that of color textures.

In addition to supervised classification applications, the discussed features are particularly attractive in image segmentation tasks. The requirements in the image segmentation applications are often diametrically opposed; the desire for a large number of image features to accurately identify uniform texture regions versus the need for a spatially compact neighbor set to allow accurate detection of texture boundaries. Based on the results of this study, the multispectral random field based approach can satisfy both requirements providing a good tool for image segmentation.

8. REFERENCES

[1] G. Paschos and M. Petrou, Histogram Ratio Color Features for Color Texture Classification, Pattern Recognition Letters, vol. 24, Jan. 2003, pp. 309-314.

[2] A. Drimbarean and P. F. Whelan, Experiments in Color Texture Analysis, Pattern Recognition Letters, vol. 22, Aug. 2001, pp. 1161-1167.

[3] G. Paschos, Fast Color Recognition Using Chromaticity Moments, Pattern Recognition Letters, vol. 21, Aug. 2000, pp. 837-841.

[4] A. Jain and G. Healey, A Multiscale Representation Including Opponent Color Features for Texture Recognition, IEEE Trans. on Image Processing, vol. 7, no. 1, Jan. 1998, pp. 124-128.

[5] B. Thai and G. Healey, Optimal Spatial Filter Selection for Illumination-Invariant Color Texture Discrimination, IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 30, no. 4, Aug. 2000, pp. 610-616.

[6] M. Mirmehdi and M. Petrou, Segmentation of Color Textures, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 2, Feb. 2000, pp. 142-159.

[7] PH Suen and G. Healey, Modeling and Classifying Color Textures Using Random Fields in a Random Environment, Pattern Recognition, vol. 32, June 1999, pp. 1009-1017.

[8] K. B. Eom, Segmentation of Monochrome and Color Textures Using Moving Average Modeling Approach, Image Vision Computing, vol. 3, no. 17, 1999, pp. 233–244.

[9] E. Oja and K. Valkealahti, Compressing Higher-Order Co-Occurrence for Texture Analysis Using the Self-Organizing Map, Proceedings of the IEEE International Conference on Neural Networks, vol. 2, Nov.-Dec. 1995, pp. 1160-1164.

[10] Vision and Modeling Group, MIT Media Laboratory, Vision Texture (VisTex) database, http://www-white.media.mit.edu/vismod/, 1995.

[11] Basic parameter values for the HDTV standard for the studio and for international programme exchange, Tech. Rep. ITU-R Recommendation BT.709, ITU, 1211 Geneva 20, Switzerland, 1990, formerly CCIR Rec. 709.

[12] J. W. Bennett, Modeling and Analysis of Gray Tone, Color, and Multispectral Texture Images by Random Field Models and Their Generalizations, Ph.D. Dissertation, Southern Methodist University, 1997.

[13] J. Bennett and A. Khotanzad, Maximum Likelihood Estimation Methods for Multispectral Random Field Image Models, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 6, June 1999, pp. 537-543.

[14] S. Haykin, Neural Networks: Multilayer Perceptrons, (1994) 138-229, Macmillan College Publishing Company, NJ.

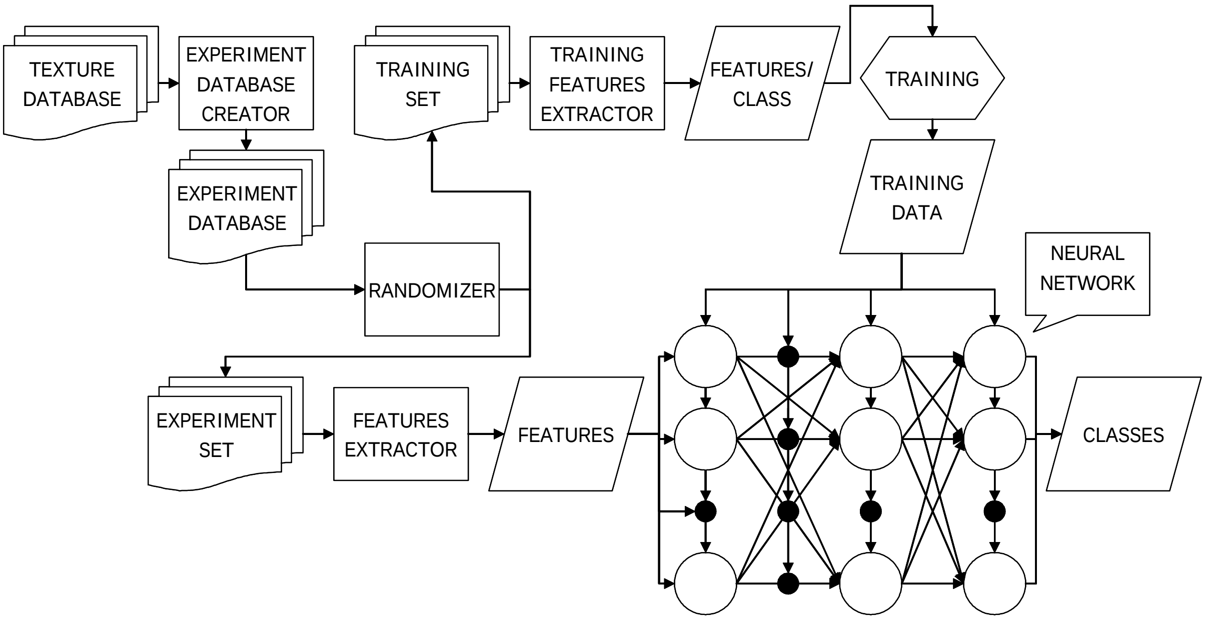
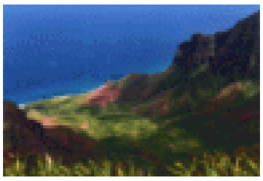


Fig. 1. Overall approach



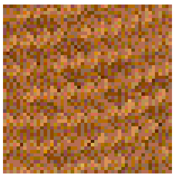


Fig. 2. Real-world texture and reference texture

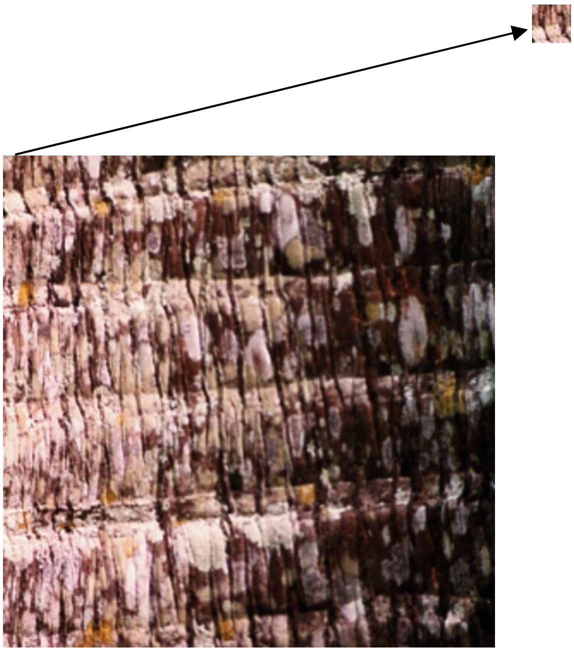


Fig. 3. Image sampling

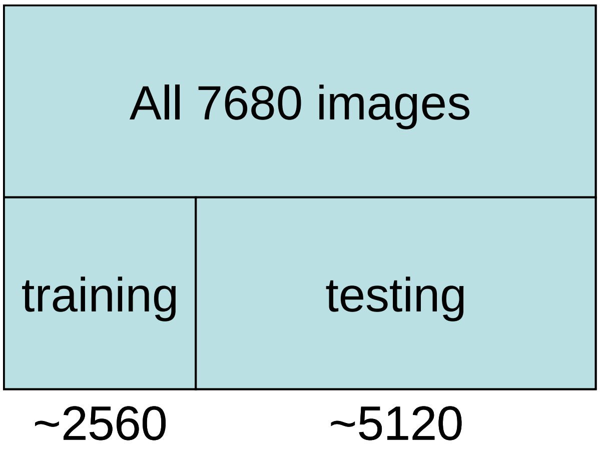


Fig. 4. Partitioning of the overall database

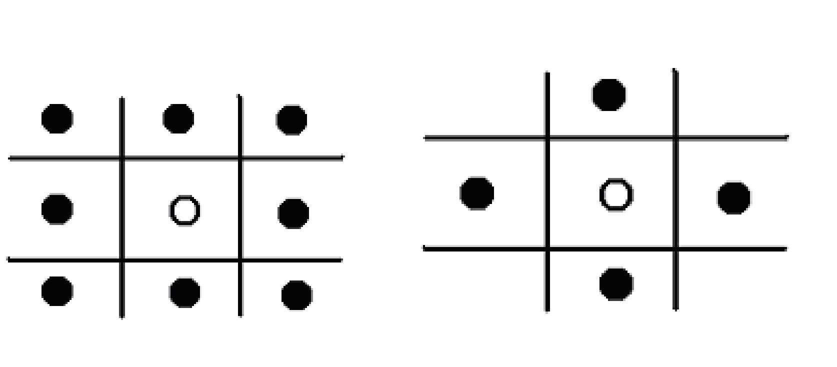


Fig. 5. Full and half neighborhood sets used in the image classification experiments. The (0,0) position is represented by the white circles, and the black circles denote the relative neighbor locations.

Table 1. Results of Feature Extraction Process from Color Texture Images

Features	Neighborhood Set	# of Features	# of Images in Database	# of Files with Features Created
MSAR	half	22	7,680	7,680
MSAR	full	40	7,680	7,680
MMRF	half	16	7,680	7,680
MMRF	full	28	7,680	7,680
PMRF	half	16	7,680	7,680
PMRF	full	28	7,680	7,680

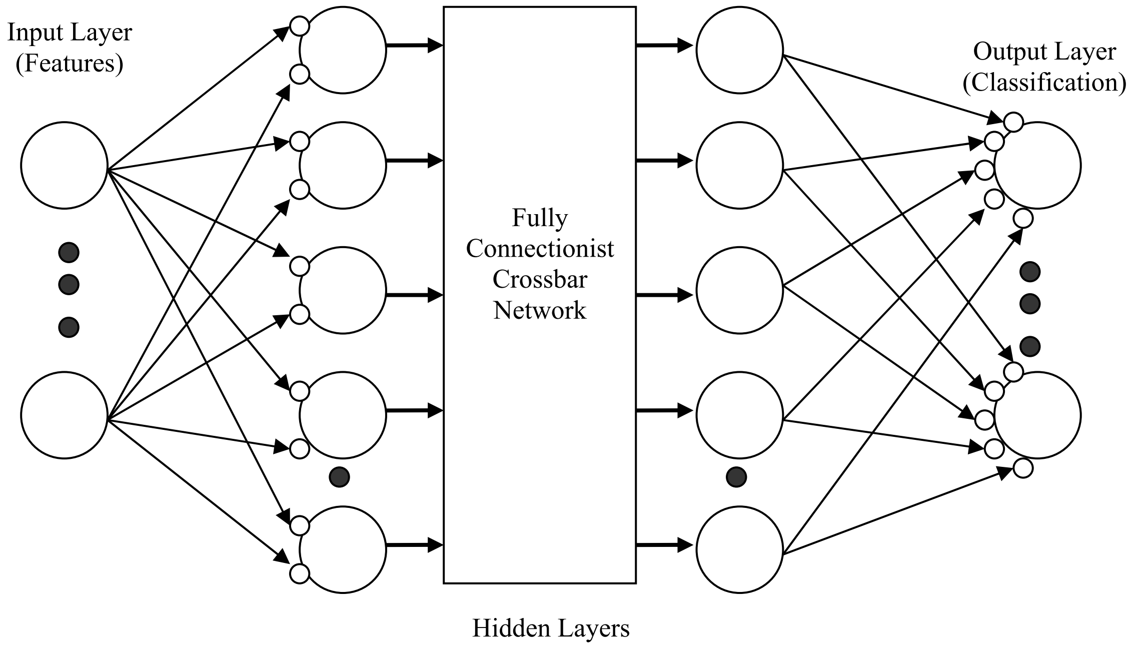


Fig. 6. This is an example of a 4 Layer fully connected neural network

Table 2. Summary of Neural Network Classification Results for 64×64 Images

Model Type	No. of Correctly Classified Out of 5,120	Accuracy Rate
Color Image Database		
MSAR (half neighborhood)	3,277	64.0%
MSAR (full neighborhood)	4,567	89.2%
MMRF (half neighborhood)	3,645	71.2%
MMRF (full neighborhood)	4,623	90.3%
PMRF (half neighborhood)	3,574	69.8%
PMRF (full neighborhood)	4,628	90.4%
Gray-Level Image Database		
GLCM (1 pixel distance)	2,299	44.9%
GLCM (2 pixels distance)	2,017	39.4%

Received: Jan. 2005. Accepted: Apr. 2005.