Unsupervised Clustering of Texture Features Using SOM and Fourier Transform

Brijesh Verma, Vallipuram Muthukkumarasamy and Changming He
School of Information Technology
Griffith University, Gold Coast Campus, QLD 9726, Australia
E-mail: {b.verma, v.muthu}@mailbox.gu.edu.au

Abstract

Texture analysis has a wide range of real-world applications. This paper presents a novel technique for texture feature extraction and compares its performance with a number of other existing techniques using a benchmark image database. The proposed feature extraction technique uses 2D-DFT transform and self-organizing map (SOM). A combination of 2D-DFT and SOM with optimal parameter settings produced very promising results. The results from large sets of experiments and detailed analysis are included in this paper.

Keywords: Texture Feature, Self-organising Map, Unsupervised Clustering, 2D-DFT

1. Introduction

As we are in an era of graphical information, we encounter millions of digital images through the World Wide Web and digital visual data sources such as digital cameras, scanners and different kinds of sensors [1]. Texture is an important attribute for image analysis, retrieval, computer vision etc. Texture analysis comprises problems like texture classification, texture segmentation, texture synthesis, and shape from texture. The proposed research will focus on texture classification. This has a large number of applications, ranging from simple content-based image retrieval, and remote sensing to medical image matching/diagnosis.

Although textured image analysis has been a topic of researcher for the last few decades [2], due to the complexity and the lack of ability to clearly define the significant features of texture, a number of challenging problems still need to be addressed. Features that have been used to describe images include simple mean and standard deviation, Gabor transforms, wavelet-based features, and Fourier transform based features [3–5]. In this paper, we propose a feature extraction technique, which uses 2D-DFT in its first stage of extraction process. Even after identifying the significant features that could represent/ identify a given image or pattern, techniques need to be developed to cluster and classify the images into well defined classes, based on those feature values. This has recently been addressed by researchers using approaches such as artificial neural networks, model and filter based approaches, etc.[6]. The existing algorithms, however, do not produce satisfactory classification rate and/or are computationally expensive.

The use of Kohonen's Self-Organising Map for clustering images based on texture features has not been fully investigated. The work presented here aims to use SOM with optimal settings for clustering images based on texture features. The developed approach is tested on a benchmark database [7].

The remainder of the paper is broken down into 3 sections. Section 2 describes the proposed research methodology, Section 3 provides experimental results and analysis, and a conclusion is drawn in Section 4.

2. Proposed Research Methodology

This section describes the proposed technique for feature extraction and classification. The Fig. 1 depicts the block diagram representation of the technique. Stage 1 deals with feature extraction from texture sub-images. Stage 2, deals with classification of features into texture classes.

The image database used in this research is the first 96 texture images in the Brodatz Texture Collection. Each of these texture images is a 512 x 512 pixels, 256 gray-scaled image, and is segmented into 16 128 x 128 pixels none-overlapping sub-images. These sub-images are transformed using the 2D- Discrete Fourier Transform (2D-DFT) [8], and the resulting magnitude matrices form the DFT Database. The 2D-DFT itself is considered as a feature extraction technique of these sub-images. But three further feature extraction techniques are performed on these magnitude matrices to extract features from them and to reduce the dimension of the feature vectors.

0-7803-7898-9/03/$17.00 ©2003 IEEE

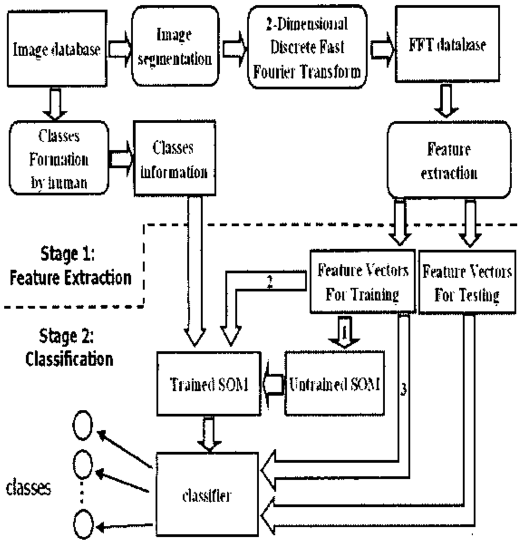
Image database
Image segmentation
2-Dimensional Discrete Fast Fourier Transform
FFT database
Classes Formation by human
Classes information
Feature extraction
Stage 1: Feature Extraction
Stage 2: Classification
Feature Vectors For Training
Feature Vectors For Testing
2
1
3
Trained SOM
Untrained SOM
classes
classifier

Figure 1: Block Diagram Of Texture Feature Extractor And Classifier

After feature extraction of the magnitude matrices, 1536 feature vectors are formed. 1152 of them are used to train the Kohonen's Self-Organizing Map (SOM) and the other 384 feature vectors are used to test the classifier. After the SOM is trained, the class information of the feature vectors is used to calibrate the SOM, and thus a texture feature classifier is created. The training and testing feature vectors are then presented to the classifier, and the classification rates of the training data and testing data are obtained.

Image Segmentation and Normalization

In order to create a number of small images which belong to the same class, each of the 512 × 512 pixels images are divided into 16 non-overlapping 128 × 128 pixels sub-images, thus forming 16 sub-images and creating a database of 1536 texture images. From every image 12 sub images are used for training and 4 sub images are used for testing, as shown in Fig 2.

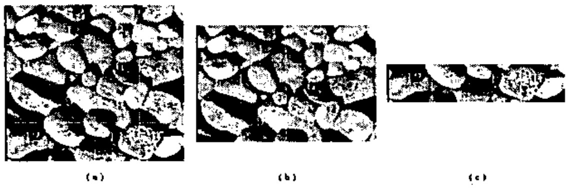

Figure 2: Training And Testing Sets From Brodatz Database
(A) Original Image (B) Training Set (C) Testing Set

After the DFT, since phase information provides little information, only magnitude coefficients matrices are used for texture characterization. All the 128x128-pixels sub images obtained from the *Image Segmentation* will be performed 2D-DFT according to equation (1). Only the 1536 magnitude matrices are used for the further feature extraction and classification processes.

$$(Ff)(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp\left(-j2\pi(\frac{ux}{M}+\frac{vy}{N})\right) \quad (1)$$

The magnitude coefficient matrix is normalized according to equation (2).

$$F'(u,v)=\frac{|F(u,v)|}{\sqrt{\sum_{u,v(u\neq 0)\vee(v\neq 0)}|F(u,v)|^2}} \quad (2)$$

3. Feature Extraction

A properly designed feature extraction method cannot only dramatically reduce the dimensions of the feature vector, but also increase the performance of the classification considerably. If 128 x 128 pixel sub-images are directly fed to a classifier, memory and computational overhead would be high. In this research, we investigate three different feature extraction techniques and compare their performance.

The first technique extracts the mean and standard deviation from the raw sub-images according to equations (3) and (4). The second technique extracts the mean and standard deviation from the magnitude matrix of the DFT transformed sub-images. Our third proposed technique considers only the upper half of the magnitude matrix. This 64 x 128 matrix is equally divided into 128 sub-matrixes as shown in Fig. 3. The size of each sub-matrix is 8 x 8. The mean values of each 8 x 8 sub-matrix is calculated thus forming a 128-dimension feature vector {mean 0, mean 1, mean 2, ..., mean 127}.

$$\mu=\frac{1}{n}\sum_{i=1}^{n}x_i \quad (3)$$

$$\sigma=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2} \quad (4)$$

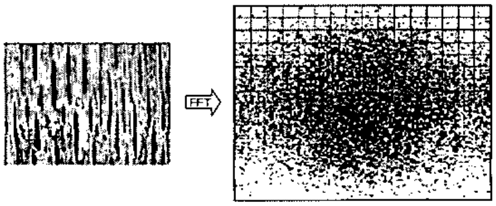

Figure 3: A Proposed Feature Extraction Technique: Mean Values Of The Sub-Matrices In Magnitude Matrices

Calibrating the Self-Organizing Map

After a number of iterations, the SOM is trained, and the reference vectors of the nodes within a certain region in the trained SOM are close to each other [9]. As the SOM is initialized with random reference vectors, it is calibrated to determine regions and to separate them. Fig. 4 shows an example of calibrating the SOM.

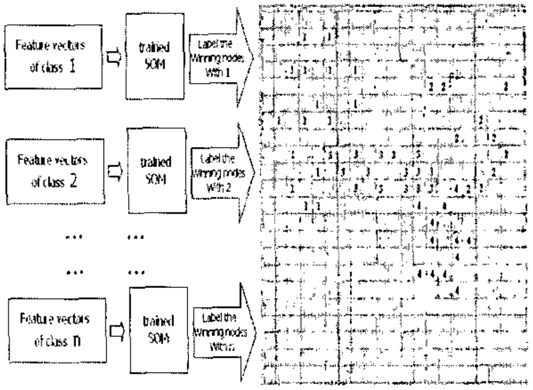
Feature vectors of class 1
trained SOM
Label the Winning nodes With 1
Feature vectors of class 2
trained SOM
Label the Winning nodes With 2
...
Feature vectors of class n
trained SOM
Label the Winning nodes With n

Figure 4: An Example Of Calibrating A SOM

Classifier For The Trained SOM

After being trained and calibrated, the SOM can now work as a classifier for the input vectors. An input vector is presented to the trained SOM, and the Euclidean Distances between the input vector and each of the nodes' reference vectors are calculated. The node with the minimum distance is called the winning node. The 8 closest labeled nodes are not treated equally in the proposed classifier. The closer a labeled node to the responding node, the more "significant" it will be. A "Significant Factor" (SF) is proposed to indicate how "significant" the labeled node *i* is to the responding node *c*.

$$SF_i = \frac{1}{dist_{ic} + 1} \qquad (5)$$

where $dist_{ic}$ is the distance between node *i* and node *c*.

3. Experimental Results

A large number of experiments were carried out to determine the optimal parameters such as significant factor, initial update radius, topology and shape of SOM. The performance of the classifier under different combinations with three feature extraction techniques was examined. Only a sample of results obtained is reported here.

Comparison of Three Feature Extraction Techniques

For the following experiments, the SOM is trained with 2000 iterations of the training data. Size of the SOM is 32 x 32, initial learning rate alpha = 0.3, initial radius = 16. "8 Neighbor Sample Points" classifier is used with the "Significant Factor" S.F.=1/(dist+1). The Table 1 compares the classification rates of the 3 feature extraction techniques.

Table 1: Comparison of Three Feature Extraction Techniques

Feature Extraction Technique	**Classification Rate for training data (%)**	**Classification Rate for testing data (%)**
Mean and Standard Deviation of each row of the original sub images	71.88	40.63
Mean and Standard Deviation of each row of the Magnitude Matrices of 2DD-FFT	86.89	72.74
Mean values of the sub-matrixes in the Magnitude Matrixes of 2DD-FFT	90.28	77.34

The Effect of the Shape of SOM on Classification Rate

The number of nodes in x direction and y direction decides the "shape" of the SOM. The number of nodes in each of the following 5 SOMs is about 400. The Classification Rates and the time to train the SOMs are compared.

For all the following experiments within this section, the SOM is trained with 1000 iterations of.the training data, alpha=0.3, radius =10, "8 Neighbor Sample Points" classifier is used, S.F.=1/(dist+1). The results are shown in Fig. 5.

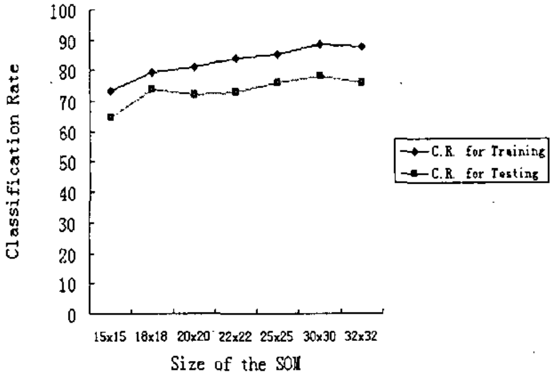

Figure 5: The Effect Of The Size Of SOM On Classification Rate

The Effect Of Initial Update Radius On Classification Rate

For the following experiments, the size of SOM is 32 x 32; initial learning rate alpha is 0.3; topology is hexagonal; number of iterations is 100. "8 Neighbor Sample Points" and S.F.=1/(dist+0.5) is used as classifier. The results are shown in Fig 6.

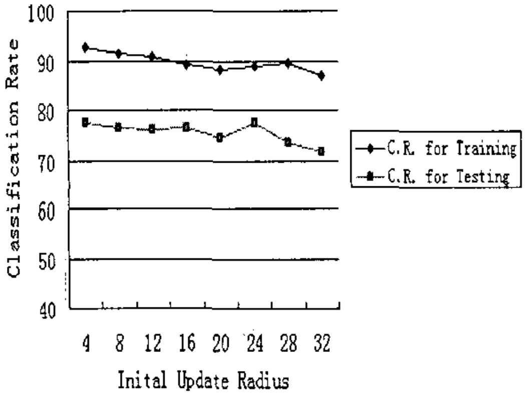

Figure 6: The Effect Of Initial Update Radius On Classification Rate

The results show that, for this specific problem, the classification rate is higher with smaller initial update radius.

The Effect Of Iteration On Classification Rate

For the following experiments, the size of the SOM is 25 x 25, alpha=0.3, radius=13. "8 Neighbor Sample Points" and S.F.=1/(dist+1) are used as the classifier. Fig. 7 shows the results obtained. In this particular experiment, as the number of iterations increase the classification rates also increase.

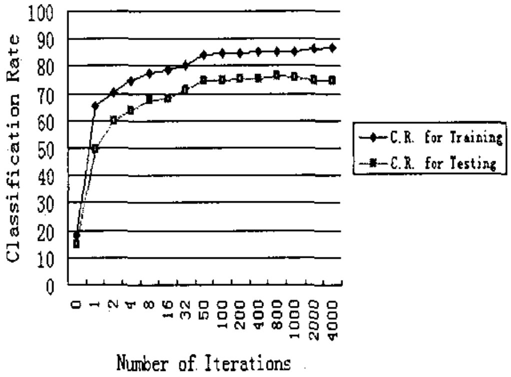

Figure 7: The Effect Of Iteration On Classification Rate

The Effect of Topology of SOM on Classification Rate

This set of experiments is designed to find out the best Topology of the SOM for this specific texture images classification problem. The SOMs are trained with 200 iterations of the training data. "8 Neighbor Sample Points", and S.F.=1/(dist+1) are used as classifier.

Table 2: The Effect Of Topology Of SOM On Classification Rate

	Rectangular			Hexagonal		
Size of SOM	C.R. For Training Data (%)	C.R. For Test Data (%)	Time (s)	C.R. For Training Data (%)	C.R. For Test Data (%)	Time (s)
32 x 32	90.4	77.3	216	91.0	77.0	229
25 x 25	87.6	76.0	130	86.7	76.8	136
20 x 20	84.8	73.1	61	82.5	71.8	65

As seen from Table 2, the experimental results show that the type of topology of the SOM has no significant effect on the Classification Rate. The time required to train a SOM with rectangular topology is slightly less than that of a SOM with Hexagonal topology.

Getting The Optimal Parameters For The Classifier

The optimal values for the number of neighbor sample points and the best significant factor are examined. The size of the SOM is 32 x 32; radius=16; alpha=0.3; topology is hexagonal. The SOM is trained with 2000 iterations of the training data. "8 Neighbor Sample Points" is used as the classifier.

The Effect of Significant Factor Function on Classification Rate

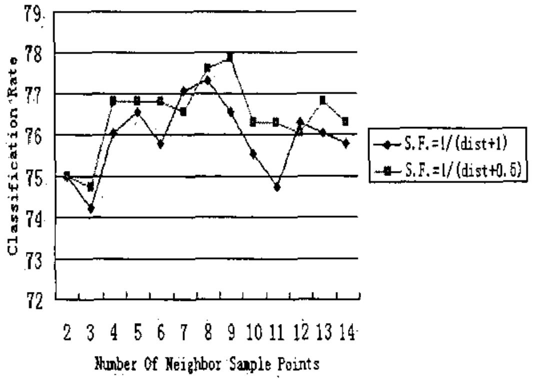

Figure 8: Comparison Of The C.R. Attained From Different Classifiers

Fig. 8 shows that the group of classifiers using function S.F.=1/(dist+0.5) perform better then their counterparts using function S.F=1/(dist+1), except two.

4. Conclusions

Texture Images Classification Using 2DD-FFT & SOM Based Techniques

We have investigated the performance of a number of feature extraction and classification techniques, including the proposed method, which uses 2D-DFT with optimal parameter settings for SOM, and Hamming network. Our proposed technique with SOM size 30 x 30, rectangle topology, number of iteration 800, initial learning rate alpha 0.3 and initial update radius 4 were the optimal parameters. For the classifier, "8 neighbor sample points" and Significant Factor Function S.F.=1/(dist+0.5) are used. The optimal classification rates obtained for training data set is 92.23%, and for the testing data set is 81.15%.

Texture Images Classification Using 2DD-FFT & Hamming Network Based Techniques

2D-DFT and Hamming network as the classifier produced classification rate of 58.20% and 49.5% for training and testing data sets respectively.

Texture Images Classification Using Auto-Associator and MLP

For comparison purposes, Auto-Associator and MLP based classifier [5] is also examined. The main idea of the auto-associator feature extractor (AAFE) is based on input:hidden:output mapping. 1152 of these feature vectors were used to train a 256:32:256 AAFE. The outputs of the hidden layer formed a 32-dimension feature vector for each of the 256-dimension feature vectors. The 1152 32-dimension feature vectors were used to train a 32:n:32 MLP.

The best classification rate obtained for training data was 86.02%. The parameters used for the AAFE and MLP were listed as follow: 256:32:256 AAFE trained with 50,000 iterations; 32:30:32 MLP trained with 100,000 iterations. The best classification rate obtained for testing data was 88.28%.

Comparison Of The Texture Images Classification Techniques

The three aforementioned techniques are compared in Fig. 9. For testing data set, AAFE & MLP gets the highest classification rate of 88.28%, followed by SOM 81.15%. Hamming network requires the least time for training, but the classification rate was the lowest of them. For training data sets, SOM performed better than AAFE & MLP. The time required to train the SOM is much less than the time required to train the AAFE and MLP.

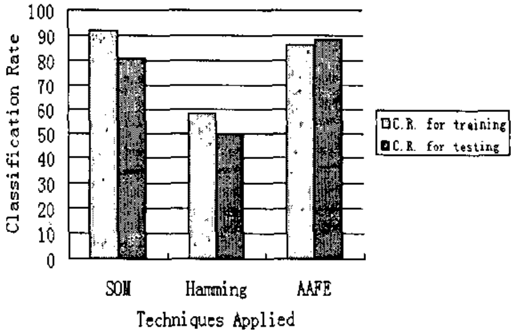

Figure 9: Comparison Of The Texture Images Classification Techniques

References

[1] S. Kulkarni, Content-based Image Retrieval using Neuro-fuzzy Techniques, Ph.D. Dissertation, Griffith University, January 2002.

[2] R.M. Haralick, K. Shanmugam and I. Distein, Textural Features for Image Classification, IEEE Trans. SMC, 6(3): 610-622, 1973

[3] W. Ma and B. Manjunath, A Comparison of Wavelet Transform Features for Texture Image Annotation, Proceedings of IEEE International Conference on Image Processing, Vol. 2, pp. 256-259, 1995.

[4] C.H. Chen, A Study of Texture Classification using Spectral Features, Proceedings of the 6th

International Conference on Pattern Recognition, pp. 1074-1077, 1982.

[5] S. Kulkarni and B. Verma, An Auto-associator for Automatic Texture Feature Extraction, Proceedings of the 4th International Conference on Computational Intelligence and Multimedia Applications, pp. 328-332, Yokosuka, Japan, 2001.

[6] A. Jain and F. Farrokhania, Unsupervised Texture Segmentation using Gabor Filters, Journal of Pattern Recognition, Vol. 24, Number 12, pp. 1167-1186, 1991.

[7] P. Brodatz, Texture: A Photographic Album for Artists and Designers, Dover Publications, New York, USA, 1966.

[8] R. Gonzales and R. Woods, Digital Image Processing, Addison-Wesley Publishing Company, 1993.

[9] T. Kohonen, Self-Organizing Maps, 3rd Extended Edition, ISBN 3-540-67921-9, 2001.