A Texture Feature Extraction Technique Using 2D–DFT and Hamming Distance

Yu Tao, Vallipuram Muthukkumarasamy, Brijesh Verma and Michael Blumenstein
School of Information Technology
Griffith University, Gold Coast Campus, QLD 9726, Australia
E-mail: {v.muthu, b.verma, m.blumenstein}@griffith.edu.au

Abstract

Texture analysis plays an increasingly important role in computer vision. Since the textural properties of images appear to carry useful information for discrimination purposes, it is important to develop significant features for texture. This paper presents a novel technique for texture extraction and classification. The proposed feature extraction technique uses 2D–DFT transformation. A combination of this technique and a Hamming Distance based neural network for classification of extracted features is investigated. The experimental results on a benchmark database and detailed analysis are presented.

1. Background

Texture analysis has a wide range of applications. Millions of digital images are created throughout the World Wide Web, digital cameras, different kinds of sensors, medical scanners etc. Image analysis is based on three main image features: colour, shape and texture. Texture plays an important role in human vision. Texture has been found to provide cues to scene depth and surface orientation. Researchers also tend to relate texture elements of varying size to a reasonable 3-D surface. Although textured image analysis has been a topic of research for the last few decades [1-12], 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 texture images include simple mean and standard deviation, Gabor transforms, wavelet-based features, and Fourier transform based features [5-11].

In this paper, we propose a feature extraction technique, which uses a 2D-Discrete Fourier Transform (2D-DFT) and investigate it in conjunction with a novel Hamming Distance based neural network to classify the texture features of the images. The proposed feature extraction technique was implemented and tested on the Brodatz benchmark database [12].

2. Research methodology

This section describes in detail the proposed technique for feature extraction and classification. The overall block diagram of texture feature extraction and classification of these features is presented in Figure 1.

The proposed technique is divided into two stages. Stage 1 deals with image segmentation and feature extraction from the texture images. Stage 2, deals with classification of features into texture classes. The texture database used to check the proposed technique consists of 96 different texture images. Each image is 512 x 512 pixels in size. The collection of Brodatz textures consists of textures of both a statistical and structural nature.

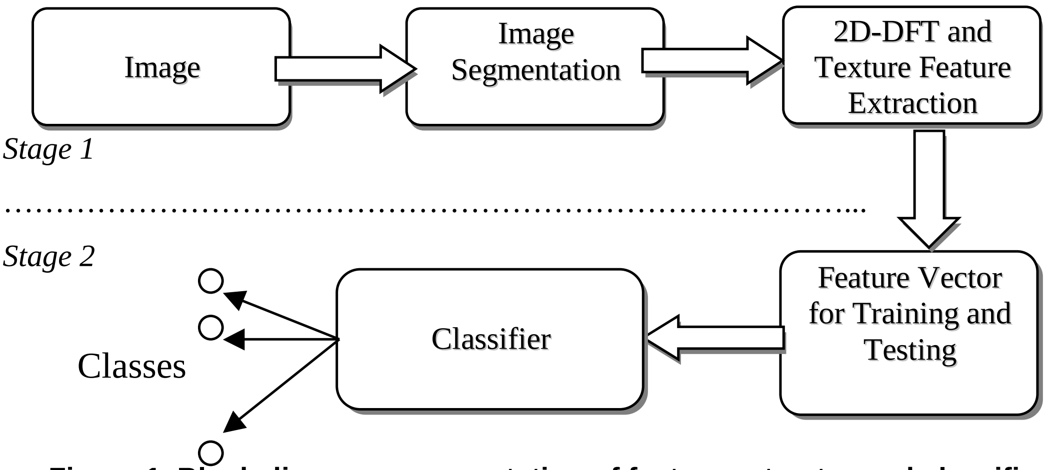

Figure 1. Block diagram representation of feature extractor and classifier

2.1 Image segmentation

The Brodatz texture database was used to evaluate the performance of the proposed techniques outlined in the previous section for texture feature extraction and classification. The database contains 96, 512 x 512 pixel texture images. In order to create a number of small images, which belong to the same class, each of the 512 x 512 images are divided into 128 x 128 sub-images, thus forming 16 sub-images from each image. The first 12 sub-images were used for training and the last 4 images were used as a testing data set, as shown in Figure 2.

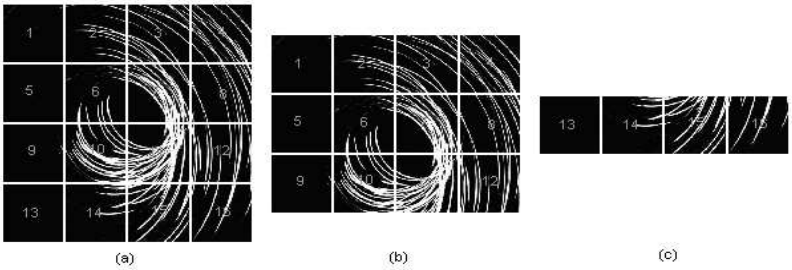

Figure 2. Extraction of training and testing sets from Brodatz database (a) Original image (b) Training set (c) Testing set

2.2 Texture feature extraction

After the images are segmented, each sub image is transformed using a 2D-DFT transformation. After the DFT, since phase values provide little information, only magnitude coefficient matrices are used for texture characterization. All the 128x128-pixel sub images obtained from image segmentation are transformed via a 2D-DFT according to equation (1). The 1536 magnitude matrices will be used for further feature extraction and classification processes.

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

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

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

2.2.1 Feature vector estimation: To reduce the size of the input vector provided to the neural network, the mean and standard deviation was calculated for each row of the sub-image (128 pixels) using the values of the pixel of each Fourier domain image. The mean and standard deviation are calculated using equations (3) and (4) respectively.

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

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

Where x is the value of the pixel in the sub-image, and n is the number of pixels - in this case it is 128. For every 128 x 128 sub-image, a 256-dimension feature vector, Xi = {mean0, s.d.0, mean1, s.d.1,….mean127, s.d.127} is derived. These feature vectors will be input to the Hamming network.

2.3 Texture feature classifier

The 96 texture patterns in the database were grouped into 32 similar classes, each of them containing 1-6 texture images. All the texture sub-images belonging to the same class were visually similar. This classification was done manually. In this research, the Hamming Distance based network is used as the texture feature classifier. Traditional *Hamming distance* is the number of bits in disagreement of two vectors. However, the Hamming distance used in our proposed classifier differs from the traditional one, mainly in two aspects: the input vectors are not presented in bits but float, and the hamming distance is not calculated in bits.

The typical Hamming network used for classification has three layers. The network contains an input layer whereby the number of nodes is equal to the number of features. It has a category layer, with as many nodes as there are categories, or classes. And finally, there is an output layer, which matches the number of nodes in the category layer.

The network is a simple feed-forward architecture with the input layer fully connected to the category layer. Each processing element in the category layer is connected back to every other element in the same layer, as well as to a direct connection to the output processing element. The output from the category layer to the output layer is done through competition.

The connection weights are first set in the input to category layer such that the matching scores generated by the outputs of the category processing elements are equal to the number of input nodes minus the Hamming distances to the example input vectors. These matching scores range from zero to the total number of input elements and are highest for those input vectors which best match the learned patterns.

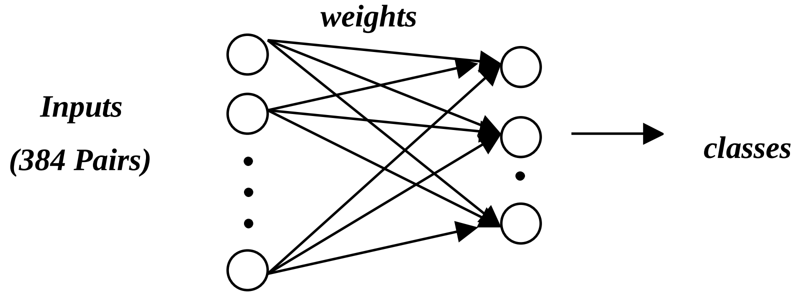

Figure 3. Hamming distance based neural network as a texture feature classier

2.3.1 Training: 1152 training pairs (*Xi)* belonging to 32 different clusters are input to the Hamming Neural Network. Every training pair is a 256-dimension vector. The weight (256 dimension vector *W*j) of every cluster is the average of the training pairs belonging to this cluster. Hamming distance = |wj – xi|;

2.3.2 Testing: There are 384 testing pairs input to the Hamming Neural Network as shown in Figure 3. Minimal Hamming distance criterion is used to classify the input to a particular class.

3. Experimental results

Three different sets of experiments are carried out to investigate the suitability of the proposed feature extraction technique and the classifier. The classification performance is tested in each of the cases.

3.1 Experiment I

For this experiment, the input images are the 2D-DFT transformed spectral images. The first 12 sub images are used as the training set and the last 4 sub images are used as the testing set, as shown in Figure 2. The total number of training pairs is 1152 and that of testing pairs is 384.

Table 1. Training and Testing Classification Rate for Experiment I

	Total # of Pairs	# of Correctly Classified Pairs	Classification Rate
Training	1152	918	79.69%
Testing	384	200	52.08%

3.2 Experiment II

In this experiment, a different extraction technique is used: the first 8 sub-images are used for training and the last 8 sub-images are used as the testing data set, as shown in Figure 4. The number of training and testing pairs is 768.

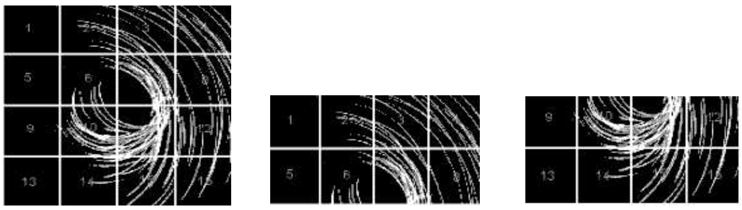

Figure 4. Extraction of training and testing sets for Experiment II
(a) Original image (b) Training set (c) Testing set

Table 2. Training and Testing Classification Rate for Experiment II

	Total # of Pairs	# of Correctly Classified Pairs	Classification Rate
Training	768	563	73.30%
Testing	768	401	52.21%

3.3 Experiment III

In this experiment, the training and testing sets are the features that are directly obtained from the sub-images, without performing a 2D-DFT transformation. The first 12 sub-images are used for training, and the last 4 sub-images are used for testing.

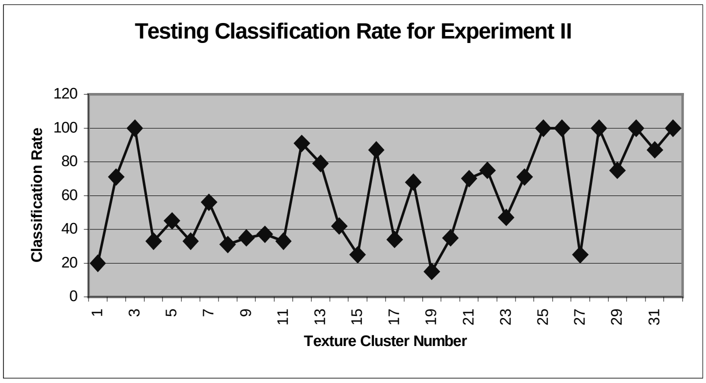

Figure 5. Testing Classification Rate for Experiment II

Table 3. Training and Testing Classification Rate for Experiment III

	Total # of Pairs	# of Correctly Classified Pairs	Classification Rate
Training	1152	411	35.67%
Testing	384	101	26.30%

4. Discussion and analysis

The results obtained from different experiments, when analysed, reveal the following outcomes.

4.1 Comparison of 2D-DFT as the texture feature extractor

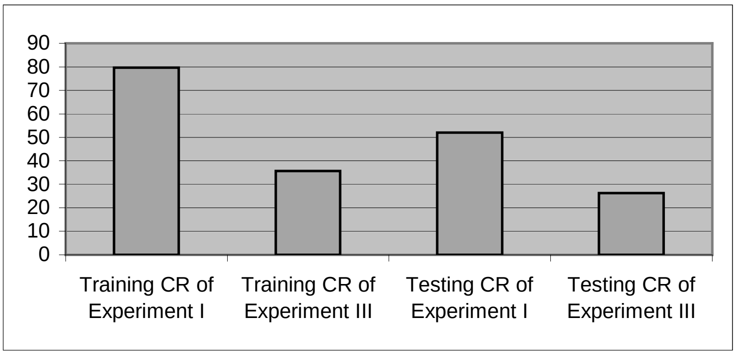

Figure 6. Comparison of Classification Result of Experiment I and III

In Experiment I, 2D-DFT is used as the texture feature extractor. The training and testing inputs to the neural network have been processed by FFT. In Experiment III, the input vector to the network has been obtained directly from the original sub-images. From the classification results shown in Figure 6, we can say that the use of 2D-DFT as a texture feature extractor has significantly improved the classification rates for both training and testing.

4.2 Analysis of the extraction of training sets and testing sets

In Experiment I, the first 12 sub-images were used for training and the last 4 images were used as the testing data set. The training and testing sets of Experiment II are different from those of Experiment I: first 8 sub images for training and last 8 for testing.

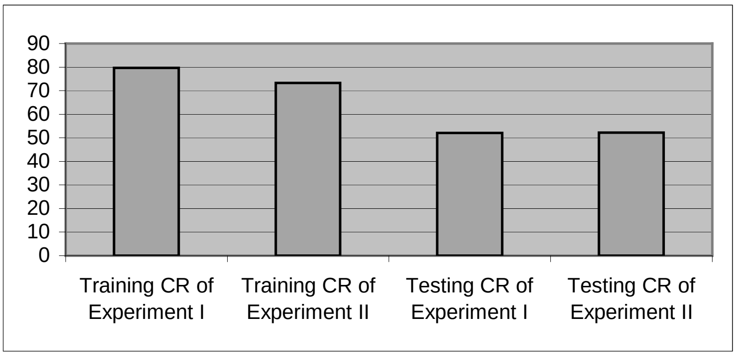

Figure 7: Comparison of Classification Results of Experiment I and II

As may be seen in Figure 7, although there is a small difference in the classification rate of the training sets, the separation of images into alternate training and testing sets does not have any significant effect on the classification rate of the testing sets.

5. Conclusions

This research has proposed and investigated a texture feature extraction technique using a 2D-DFT and a hamming distance based neural classifier. The experiments were conducted using the Brodatz album benchmark database. The results obtained are very promising and showed that the proposed 2D-DFT based feature extractor has improved the classification rate significantly. The classification rate using the proposed technique based on 2D-DFT is approximately 26% higher than that of the algorithm without using 2D-DFT.

As may be seen from the results obtained, the proposed texture feature extraction technique is very promising, however the hamming distance based network classifier in its current form, which was investigated in this research, did not produce significant results. Thus, further investigation needs to be carried out to determine a more suitable classifier architecture.

6. References

[1] B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 8, No. 18, 1996, pp. 837-842.

[2] W. Niblack, "The QBIC Project: Querying Images by Content using Color, Texture and Shape", In SPIE Proceedings of Storage and Retrieval for Color and Image Video Databases, 1993, pp. 173-187.

[3] A. K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters", Pattern Recognition, Vol. 24, No. 12, 1991, pp. 1167-1186.

[4] Y. Rubner and C. Tomasi, "Texture Matrices". Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San-Diego, USA, 1998, pp. 4601-4607.

[5] F. Lui and R. Picard. "Periodicity, Directionality and Randomness: Wold Features for Image Modelling and Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No.7, 1996, pp. 722-733.

[6] 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, Yokosuka, Japan, 2001, pp. 328-332.

[7] W. Y. Ma and B. S. Manjunath. "Texture Features and Learning Similarity", Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, USA, 1996.

[8] D. Jones and P. Jackway. "Using Granold for Texture Classification", Fifth International Conference on Digital Image Computing, Techniques and Applications, DICT 99, Perth, Australia, 1999, pp. 270-274.

[9] B. Lerner, H. Guterman, M. Aladjem and H. Dinstein. "A Comparative Study of Neural Network based Feature Extraction Paradigms", Pattern Recognition Letters, Vol. 20, 1999, pp. 7-14.

[10] D. Jones, and P. Jackway. "Using Granold for Texture Classification". Fifth International Conference on Digital Image Computing, Techniques and Applications Perth, Australia, 1999, pp. 270-274.

[11] L.Wang and J. Liu. "Texture Classification using Multiresolution Markov Random Field Models", Pattern Recognition Letters, vol. 20, 1999, pp. 171-182.

[12] P. Brodatz, "Texture: A Photographic Album for Artists and Designers", Dover Publications, New York, USA, 1996.