

Texture Classification Based on DCT and Soft Computing

Golam Sorwar, Ajith Abraham and Laurence S Dooley

School of Computing and Information Technology

Monash University (Gippsland Campus), Churchill, Victoria 3842, Australia

Email: {Golam.Sorwar, Ajith.Abraham, Laurence.Dooley}@infotech.monash.edu.au

Abstract

Classification of texture pattern is one of the most important problems in pattern recognition. In this paper, we present a classification method based on the Discrete Cosine Transform (DCT) coefficients of texture image. As the DCT works on gray level image, the color scheme of each image is transformed into gray levels. For classifying the images with DCT, we used two popular soft computing techniques namely neurocomputing and neuro-fuzzy computing. We used a feedforward neural network trained by backpropagation algorithm and an evolving fuzzy neural network to classify the textures. The soft computing models were trained using 80% of the texture data and remaining was used for testing and validation purposes. A performance comparison was made among the soft computing models for the texture classification problem. We also analyzed the effects of prolonged training of neural networks. It is observed that the proposed neuro-fuzzy model performed better than neural network.

I. INTRODUCTION

Texture as a primitive visual cue has been studied for a long time. Various techniques have been developed for texture segmentation, classification and synthesis. Although texture analysis has a long history, its applications to real image data have been limited to-date. An important and emerging application where texture analysis can make a significant contribution is the area of content-based retrieval in large image and video databases. Using texture as a visual feature, one can query a database to retrieve similar patterns. Texture classification and segmentation schemes are very important in answering such queries.

Most of the existing approaches for texture feature extraction make use of statistical techniques. Processing the texture image data requires large storage space and computational load to calculate the feature's matrix such as the SGLDM (Spatial Gray Level Dependence Matrix) [1]. In spite of the large size of SGLDM, the set of their scalar features calculated from the matrix is not efficient to represent the characteristics of image content. Another popular approach for texture classification is to use multi-resolution techniques such as wavelet transforms [12][13][14]. Genetic programming has also been used for adaptation of a 2D-lookup algorithm for blind texture detection [11].

In general, neighboring pixels within an image tend to be highly correlated. As such, it is desirable to use an invertible transform to concentrate randomness into fewer, decorrelated

parameters. The DCT technique has been shown to be near optimal for a large class of images with energy concentration and decorrelated parameters. It has been also adopted as the JPEG and MPEG coding standard [2][3]. The DCT decomposes the signal into underlying spatial frequencies, which then allows further processing techniques without sacrificing the precision of DCT coefficients or affecting the Human Visual System (HVS) model. The DCT coefficients of an image tend themselves as a new feature, which have the ability to represent the regularity, complexity and some texture features of an image [4], which can be directly applied to image data in the compressed domain. Perhaps DCT might be the right technique to handle the large memory space and the computational complexity of the existing methods.

Soft computing was first proposed by Zadeh [9] to construct new generation computationally intelligent hybrid systems consisting of neural networks, fuzzy inference system, approximate reasoning and derivative free optimization techniques. It is well known that the intelligent systems, which can provide human like expertise such as domain knowledge, uncertainty reasoning, and adaptation to a noisy and time varying environment, are important in tackling practical computing problems. In contrast with conventional artificial intelligence techniques which only deal with precision, certainty and rigor the guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, low solution cost, robustness, partial truth to achieve tractability, and better rapport with reality.

In our research, an artificial neural network is trained using backpropagation algorithm and an integrated neuro-fuzzy model for classifying the texture information based on DCT. The soft computing models were evaluated based on their classification efficiency of the different texture data sets. We also evaluated the performance of the neural network by increasing the training epochs. In section II, we present DCT transform followed by texture feature extraction in section III. In Section IV and V we present some basic theoretical aspects of neural networks and neuro-fuzzy systems followed by experimentation set up in section VI. Some discussions and conclusions are provided towards the end.

II. BLOCK DCT BASED TRANSFORM

For the DCT transform, we have to convert the RGB image into gray level image. For spatial localization, we then use the block-based DCT transformation. Each image is divided into

$N*N$ sized sub-blocks. The two dimensional DCT can be written in terms of pixel values $f(i, j)$ for $i, j = 0, 1, \dots, N-1$ and the frequency-domain transform coefficients $F(u, v)$:

$$F(u, v) = \frac{1}{\sqrt{2N}} C(u)C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \times \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (1)$$

for $u, v = 0, 1, \dots, N-1$

where

$$c(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } x = 0 \\ 1 & \text{otherwise} \end{cases}$$

The inverse DCT transform is given by

$$f(i, j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)F(u, v) \times \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (2)$$

for $i, j = 0, 1, \dots, N-1$.

III. TEXTURE FEATURE VECTOR EXTRACTION

For efficient texture feature extraction, we used the DCT coefficients in compressed domain. Each sub block contains one DC coefficient and other AC coefficients as shown in Figure 1. Since it is well known that the HVS is less sensitive to errors in high frequency coefficients while compared to lower frequency coefficients, our extracted feature sets consisted of 9 vector components (DC coefficient represents each sub-block's average energy or intensity of the block and remaining 8 coefficients representing the different pattern of image variation and directional information of the texture). The coefficients of the upper most region and those of the lower most region in a DCT transform domain represent some vertical and horizontal edge information, respectively as shown in Fig. 1(c).

IV. ARTIFICIAL NEURAL NETWORKS

Neural networks are computer algorithms inspired by the way information is processed in the nervous system [5]. An important difference between neural networks and other AI techniques is their ability to learn. The network "learns" by adjusting the interconnections between layers. When the network is adequately trained, it is able to generalize relevant output for a set of input data. A valuable property of neural networks is that of generalisation, whereby a trained neural network is able to provide a correct matching in the form of output data for a set of previously unseen input data.

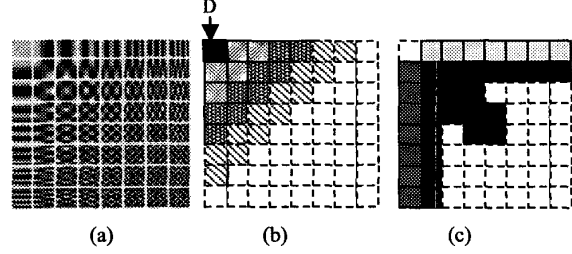


Fig.1. (a) DCT basic pattern (b) Vector element from frequency components (c) Vector elements from directional information.

Learning typically occurs by example through training, where the training algorithm iteratively adjusts the connection weights (synapses). Backpropagation (BP) is one of the most famous training algorithms for multilayer perceptrons. BP is a gradient descent technique to minimize the error E for a particular training pattern. For adjusting the weight (w_{ij}) from the i -th input unit to the j -th output, in the batched mode variant the descent is based on the gradient $\nabla E \left(\frac{\delta E}{\delta w_{ij}} \right)$ for the total training set:

$$w_{ij}(n) = -\epsilon * \frac{\delta E}{\delta w_{ij}} + \alpha * w_{ij}(n-1) \quad (3)$$

The gradient gives the direction of error E . The parameters ϵ and α are the learning rate and momentum respectively [6].

V. NEURO-FUZZY SYSTEMS

A neuro-fuzzy system [7] is defined as a combination of ANN and Fuzzy Inference System (FIS) in such a way that neural network learning algorithms are used to determine the parameters of FIS. An even more important aspect is that the system should always be interpretable in terms of fuzzy *if-then* rules, because it is based on the fuzzy system reflecting incomplete knowledge. We used the Evolving Fuzzy Neural Network [8] implementing a Mamdani type FIS. EFuNN have a five-layer architecture. The input layer represents input variables. The second layer of nodes represents fuzzy quantification of each input variable space. Each input variable is represented here by a group of spatially arranged neurons to represent a fuzzy quantization of this variable. Different membership functions can be attached to these neurons (triangular, Gaussian, etc.). The nodes representing membership functions can be modified during learning. The third layer contains rule nodes that evolve through hybrid supervised/unsupervised learning. The rule nodes represent prototypes of input-output data associations, graphically represented as an association of hyper-spheres from the fuzzy input and fuzzy output spaces. Each rule node r is defined by two vectors of connection weights - $W_1(r)$ and $W_2(r)$, the latter being adjusted through supervised learning based on the output error, and the former being adjusted through unsupervised learning based on similarity measure within a local area of the input problem space. The fourth layer of neurons represents fuzzy quantification for the output variables. The fifth layer represents

the real values for the output variables. EFuNN evolving algorithm used in our experimentation was adapted from [8].

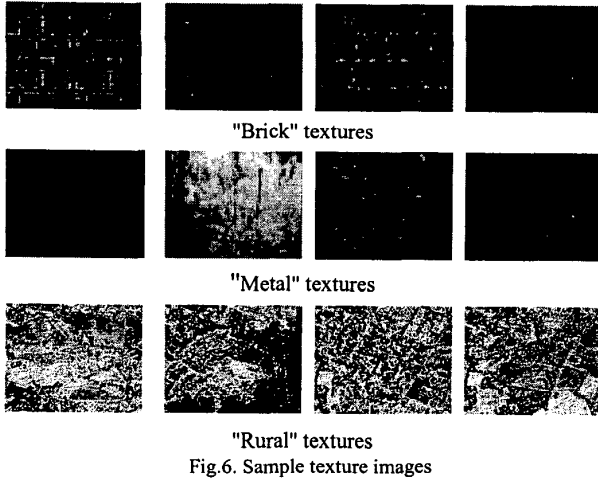


Fig. 6. Sample texture images

VI. EXPERIMENTATION SETUP AND RESULTS

In this paper, we attempt to classify 3 different types of textures using soft computing techniques. We used DCT coefficients to represent the different textures. Each texture image was represented by 327 DCT coefficients. Our texture database consisted of 240 different textures and they were manually classified into three different classes (*brick*, *metal* and *rural*). A set of sample texture images representing each class is illustrated in Figure 6. For training the soft computing models, we used 192 datasets and remaining 48 texture datasets were used for testing purposes. While the proposed neuro-fuzzy model was capable of determining the architecture automatically, we had to do some initial experiments to determine the architecture (number of hidden neurons and number of layers) of the neural network. After a trial and error approach we found that the neural network was giving good generalization performance when we had 2 hidden layers with 90 neurons each. In the following sections we report the details of our experimentations with neural networks and neuro-fuzzy models. Experiments were repeated three times and the worst errors were reported.

Neuro-Fuzzy Network Training

We used 4 Gaussian membership functions for each input variable and the following evolving parameters: sensitivity threshold $Sthr=0.99$, error threshold $Errthr=0.001$. EFuNN training has created 162 rule nodes. Empirical results are reported in Table 1 and 2.

Neural Network Training

We used a neural network trained using backpropagation algorithm. We used 1 input layer, 2 hidden layers and an output layer [327-90-90-3]. Input layer consists of 327 neurons corresponding to the input variables. The first and second hidden

layer consists of 90 neurons each with tanh-sigmoidal activation function. The initial learning rate and momentum were set as 0.05 and 0.1 respectively. Training errors (RMSE) and performance achieved after 5000, 15,000, 20,000 and 40,000 epochs are reported in Table 1 and 2. Approximate computational load in Giga-flops is reported in Table 1 and is graphically depicted in Figure 7.

TABLE I
TRAINING PERFORMANCE OF SOFT COMPUTING MODELS

	EFuNN	ANN (5000 ep)	ANN (15000 ep)	ANN (20000 ep)	ANN (40,000 ep)
RMSE (Training)	0.2e ⁻⁰⁰³	3.9e ⁻⁰⁰³	1.4e ⁻⁰⁰⁴	9.4e ⁻⁰⁰⁵	8.5e ⁻⁰⁰⁵
Giga Flops	152	187860	751100	939000	1502200

TABLE II
TEST RESULTS FOR TEXTURE CLASSIFICATION

	Neuro-Fuzzy	Artificial Neural Network				
		No of epochs				
		5000	15000	20000	40000	
Brick (16 Nos)	A ₁	15	14	14	12	12
	A ₂	1	2	2	4	4
Metal (16 Nos)	B ₁	16	13	14	15	15
	B ₂	0	3	2	1	1
Rural (16 Nos)	C ₁	11	12	11	14	14
	C ₂	5	4	5	2	2
Total (48 Nos)	X = (A ₁ +B ₁ +C ₁)	42	39	39	41	41
	Y = (A ₂ +B ₂ +C ₂)	6	9	9	7	7
*Reliability of classification (%)		88	81.25	81.25	85.4	85.4

$$*Reliability = \left(\frac{x}{48} \right) * 100$$

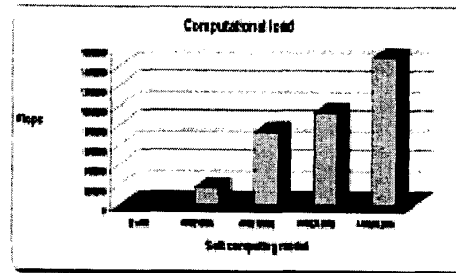


Fig. 7. Computational load of soft computing models

Test results

Table 2 summarizes the comparative performance of EFuNN and ANN. The best classification was obtained using EFuNN (88%) and was 85.4% for neural networks.

VII. CONCLUSIONS

In this paper, we attempted to classify 3 different types of textures using artificial neural networks and a neuro-fuzzy system. The texture features were represented by DCT coefficients, which do not require additional complex computation for feature extraction. As the high frequency coefficient is less sensitive to human visual systems, we constructed a feature matrix considering the first few coefficients of each block. The proposed neuro-fuzzy model (EFuNN) performed well with respect to the neural network trained using backpropagation algorithm. As depicted in Table 1, neuro-fuzzy model is less computational expensive when compared to neural networks. As EFuNN adopts a one-pass (one epoch) training technique, it is highly suitable for online learning. Hence online training can incorporate further knowledge very easily. While neural network is often called a *black-box*, an important advantage of neuro-fuzzy model is its reasoning ability (*if-then* rules) of any particular state [10]. We also studied the generalization performance of BP training when the training epochs were increased from 5000 epochs to 40,000 epochs. When the number of epochs were increased, it was interesting to note the continuous reduction of the training error (RMSE) but the generalization error (classification accuracy) however tends to settle after 20,000 epochs. The *rural* texture classification, neural network after 20,000 epochs training performed better than the proposed neuro-fuzzy model.

The proposed prediction models based on soft computing on the other hand are easy to implement and produces desirable mapping function by training on the given data set. Moreover, the considered connectionist models are very robust, capable of handling noisy and approximate data and might be more reliable in worst situations. Choosing suitable parameters for the soft computing models is more or less a trial and error approach. Optimal results will depend on the selection of parameters. Selection of optimal parameters may be formulated as an evolutionary search to make connectionist models fully adaptable and optimal according to the requirement. Our future research on texture retrieval will be oriented in this direction.

REFERENCES

- [1] F. Borko, video and image processing in multimedia systems, Kluwer Academic publishers, 1995, pp 225-249.
- [2] G.K. Wallace, " Overview of the JPEG still image compression standard," SPIE, Vol. 1244, 1990, pp. 220-233.
- [3] D.J. Le Gall, " The MPEG video compression algorithm: a review," SPIE Vol. 1452, 1991, pp. 444-457.
- [4] S.M. Lee, Hee_Jung Bae, and Sung-Hwan Jung, "Efficient content-based image retrieval methods using color and texture," ETRI Journal, Vol. 20, pp. 272-283, 1998.
- [5] J.M. Zurada, Introduction to artificial neural systems, PWS Pub Co, 1992.
- [6] A. Abraham and B. Nath, Optimal design of neural nets using hybrid algorithms, In proceedings of 6th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000), pp. 510-520, 2000.
- [7] A. Abraham and B. Nath, Designing optimal neuro-fuzzy systems for intelligent control, The Sixth International Conference on Control, Automation, Robotics and Vision, (ICARCV 2000), December 2000.
- [8] N. Kasabov, Evolving fuzzy neural networks - algorithms, applications and biological motivation, in Yamakawa T and Matsumoto G (Eds), Methodologies for the Conception, Design and Application of Soft Computing, World Scientific, pp. 271-274, 1998.
- [9] L.A. Zadeh, Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems, Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, O Kaynak, LA Zadeh, B Turksen, IJ Rudas (Eds.), pp1-9, 1998.
- [10] N. Kasabov and B. Woodford,. Rule insertion and rule extraction from evolving fuzzy neural networks: algorithms and applications for building adaptive, intelligent expert systems, In Proceedings of the FUZZ-IEEE'99 International Conference on Fuzzy Systems, Seoul, Korea, 1999.
- [11] M. Koppen and X. Liu, Proceedings of the 2001 Congress on Evolutionary Computation, 2001, Volume: 2, pp: 867-872, 2001.
- [12] M.N. Shirazi, H. Noda and N. Takao, Texture modeling and classification in wavelet feature space. In Proceedings of the International Conference on Image Processing, Vol.1, pp.272-5, 2000.
- [13] I. Daubechies, The wavelet transforms, time-frequency localization and signal analysis. IEEE Trans. on Information Theory, vol.36, no.5, pp.961-1005, 1990.
- [14] J.R. Smith and S. Chang, Transform features for texture classification and discrimination in large image databascs. In Proceedings of ICIP-94, vol. 3, pp.407-11, 1994.