

A MATLAB based Face Recognition System using Image Processing and Neural Networks

Jawad Nagi, Syed Khaleel Ahmed
Department of Electrical and Electronics Engineering
Universiti Tenaga Nasional
Km7, Jalan Kajang-Puchong, 43009 Kajang, Malaysia
jawad@uniten.edu.my, syedkhaleel@uniten.edu.my

Farrukh Nagi
Department of Mechanical Engineering
Universiti Tenaga Nasional
Km7, Jalan Kajang-Puchong, 43009 Kajang, Malaysia
farrukh@uniten.edu.my

Abstract—Automatic recognition of people is a challenging problem which has received much attention during recent years due to its many applications in different fields. Face recognition is one of those challenging problems and up to date, there is no technique that provides a robust solution to all situations. This paper presents a new technique for human face recognition. This technique uses an image-based approach towards artificial intelligence by removing redundant data from face images through image compression using the two-dimensional discrete cosine transform (2D-DCT). The DCT extracts features from face images based on skin color. Feature-vectors are constructed by computing DCT coefficients. A self-organizing map (SOM) using an unsupervised learning technique is used to classify DCT-based feature vectors into groups to identify if the subject in the input image is “present” or “not present” in the image database. Face recognition with SOM is carried out by classifying intensity values of grayscale pixels into different groups. Evaluation was performed in MATLAB using an image database of 25 face images, containing five subjects and each subject having 5 images with different facial expressions. After training for approximately 850 epochs the system achieved a recognition rate of 81.36% for 10 consecutive trials. The main advantage of this technique is its high-speed processing capability and low computational requirements, in terms of both speed and memory utilization.

Keywords—Face recognition, discrete cosine transform, self-organizing map, neural network, artificial intelligence.

I. INTRODUCTION

FACE recognition has become a very active area of research in recent years mainly due to increasing security demands and its potential commercial and law enforcement applications. The last decade has shown dramatic progress in this area, with emphasis on such applications as human-computer interaction (HCI), biometric analysis, content-based coding of images and videos, and surveillance[2]. Although a trivial task for the human brain, face recognition has proved to be extremely difficult to imitate artificially, since although commonalities do exist between

faces, they vary considerably in terms of age, skin, color and gender. The problem is further complicated by differing image qualities, facial expressions, facial furniture, background, and illumination conditions[3]. A generic representation of a face recognition system is shown in Fig. 1.

This paper presents a novel approach for face recognition that derives from an idea suggested by Hjelmås and Low[1]. In their survey, they describe a preprocessing step that attempts to identify pixels associated with skin independently of face-related features. This approach represents a dramatic reduction in computational requirements over previous methods.

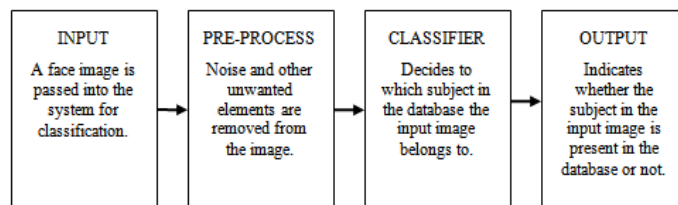


Fig. 1. Generic representation of a face recognition system

Since skin color in humans varies by individual, research has revealed that intensity rather than chrominance is the main distinguishing characteristic. The recognition stage typically uses an intensity (grayscale) representation of the image compressed by the 2D-DCT for further processing[2]. This grayscale version contains intensity values for skin pixels.

A block diagram of the proposed technique of the face recognition system is presented in Fig. 2. In the first stage, the 2D-DCT for each face image is computed, and feature vectors are formed from the discrete cosine transform (DCT) coefficients. The second stage uses a self-organizing map (SOM) with an unsupervised learning technique to classify vectors into groups to recognize if the subject in the input image is “present” or “not present” in the image database. If the subject is classified as present, the best match image found in the training database is displayed as the result, else the result displays that the subject is not found in the image database.

The rest of this paper is organized as follows: Section II discusses DCT computation on face images. Section III describes the design and architecture of the SOM neural network. Section IV shows experimental results, and discusses possible modifications and improvements to the system. Section V presents concluding remarks.

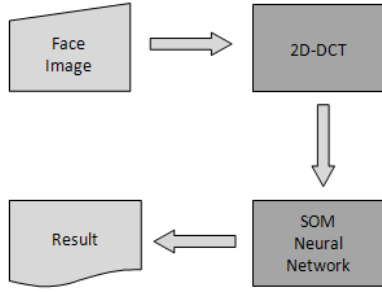


Fig. 2. Proposed technique for face recognition system

II. DISCRETE COSINE TRANSFORM

A. Overview

The discrete cosine transform is an algorithm widely used in different applications. The most popular use of the DCT is for data compression, as it forms the basis for the international standard loss image compression algorithm known as JPEG[5]. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients. Extracted DCT coefficients can be used as a type of signature that is useful for recognition tasks, such as face recognition[6,7].

Face images have high correlation and redundant information which causes computational burden in terms of processing speed and memory utilization. The DCT transforms images from the spatial domain to the frequency domain. Since lower frequencies are more visually significant in an image than higher frequencies, the DCT discards high-frequency coefficients and quantizes the remaining coefficients. This reduces data volume without sacrificing too much image quality[3]. The 2D-DCT of an $M \times N$ matrix A is defined as follows:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\left(\frac{\pi(2m+1)p}{2M}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right), \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix} \quad (2.1)$$

The values B_{pq} are the *DCT coefficients*. The DCT is an invertible transform, and the 2D-IDCT (2D Inverse-DCT) is defined as follows:

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos\left(\frac{\pi(2m+1)p}{2M}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right), \quad \begin{matrix} 0 \leq m \leq M-1 \\ 0 \leq n \leq N-1 \end{matrix} \quad (2.2)$$

The values α_p and α_q in (2.1) and (2.2) are given by:

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{M}}, & p=0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad \alpha_q = \begin{cases} \sqrt{\frac{1}{N}}, & q=0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases} \quad (2.3)$$

The proposed technique uses the *DCT transform matrix* in the MATLAB Image Processing Toolbox. This technique is efficient for small square inputs such as image blocks of 8×8 pixels. The $M \times M$ transform matrix T is given by:

$$T_{pq} = \begin{cases} \sqrt{\frac{1}{M}} & p=0, \quad 0 \leq q \leq M-1 \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2q+1)p}{2M} & 1 \leq p \leq M-1, \quad 0 \leq q \leq M-1 \end{cases} \quad (2.4)$$

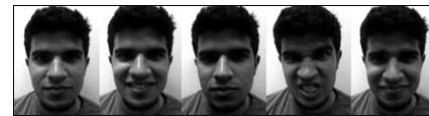
B. Face Image Preprocessing

Face images of different candidates with different facial expressions are taken with a Canon Powershot S3 IS 6.0 megapixel digital camera in the size of 1200×1600 pixels (2.0 megapixels). All face images taken resemble the following general features:

- Uniform illumination conditions
- Light color background
- Faces in upright and frontal position
- Tolerance for tilting and rotation up to 20 degrees



(a)



(b)

Fig. 3. Face images of candidates. (a) Face images of different subjects.

(b) Face image of a single subject with 5 different facial expressions

Face images are preprocessed in Adobe Photoshop CS2. The face image fabrication process is shown in Fig. 4. Image preprocessing includes the following steps:

- Auto adjusting hue and saturation levels
- Adjusting brightness and contrast to fixed scale
- Desaturating 24 bit RGB color into 8 bit grayscale
- Downsizing images to 512×512 pixels
- Saving images in jpeg format

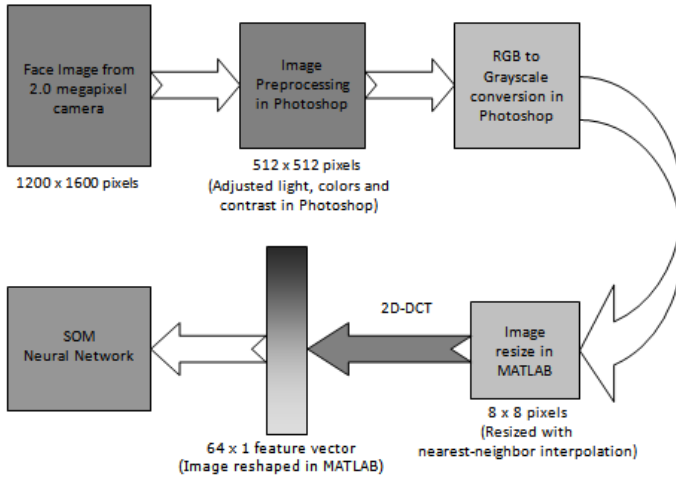


Fig. 4. Face image fabrication process

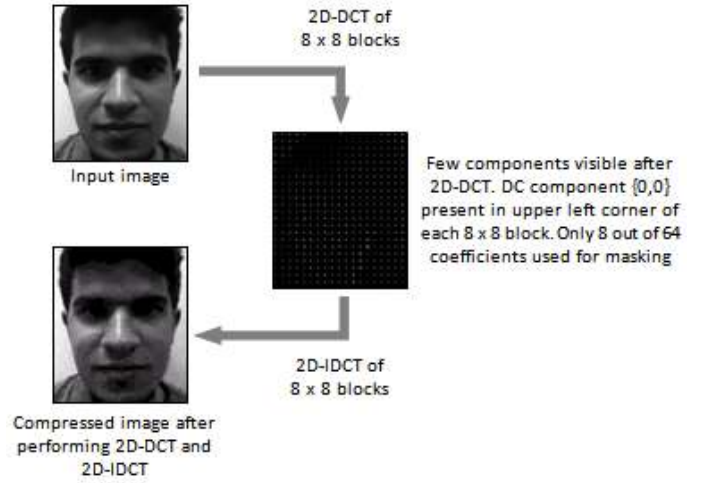


Fig. 5. 2D-DCT computation of face image

C. 2D-DCT Image Compression

Nearest-neighbor interpolation is performed using the MATLAB Image Processing Toolbox to resize preprocessed images from size 512×512 pixels to image blocks of size 8×8 pixels as shown in Fig. 4.

The proposed design technique calculates the 2D-DCT of the image blocks of size 8×8 pixels using '8' out of the 64 DCT coefficients for masking. The other 56 remaining coefficients are discarded (set to zero). The image is then reconstructed by computing the 2D-IDCT of each block using the *DCT transform matrix* computation method. Finally, the output is a set of arrays. Each array is of size 8×8 pixels and represents a single image. These steps are represented in Fig. 5 for a sample image. Empirically, the upper left corner of each 2D-DCT matrix contains the most important values, because they correspond to low-frequency components within the processed image block[2].

III. SELF-ORGANIZING MAPS

A. Overview

The self-organizing map also known as a *Kohonen Map* is a well-known artificial neural network. It is an unsupervised learning process, which learns the distribution of a set of patterns without any class information. It has the property of topology preservation. There is a competition among the neurons to be activated or fired. The result is that only one neuron that wins the competition is fired and is called the "winner"[4]. A SOM network identifies a winning neuron using the same procedure as employed by a competitive layer. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the *Kohonen Rule*. The Kohonen rule allows the weights of a neuron to learn an input vector, and because of this it is useful in recognition applications. Hence, in this system, a SOM is employed to classify DCT-based vectors into groups to identify if the subject in the input image is "present" or "not present" in the image database[3].

B. Network Architecture

SOMs can be one-dimensional, two-dimensional or multi-dimensional maps. The number of input connections in a SOM network depends on the number of attributes to be used in the classification[4].

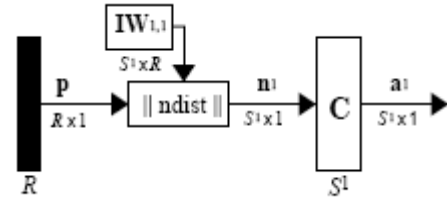


Fig. 6. Architecture of a simple SOM

The input vector \mathbf{p} shown in Fig. 6 is the row of pixels of the DCT compressed image. The $\|\text{dist}\|$ box accepts the input vector \mathbf{p} and the input weight matrix $\mathbf{IW}_{1,1}^{1,1}$, which produces a vector having S^1 elements. The elements are the negative of the distances between the input vector and vectors $\mathbf{IW}_{1,1}^{1,1}$ formed from the rows of the input weight matrix. The $\|\text{dist}\|$ box computes the net input \mathbf{n}^1 of a competitive layer by finding the Euclidean distance between input vector \mathbf{p} and the weight vectors. The competitive transfer function \mathbf{C} accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the *winner*, the neuron associated with the most positive element of net input \mathbf{n}^1 . The winner's output is 1. The neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1. Thus the competitive transfer function \mathbf{C} produces a 1 for output element \mathbf{a}^1_i corresponding to i^* , the "winner". All other output elements in \mathbf{a}^1 are 0[8].

$$\mathbf{n}^1 = -\|\mathbf{IW}_{1,1} - \mathbf{p}\| \quad (3.1)$$

$$\mathbf{a}^1 = \text{compet}(\mathbf{n}^1) \quad (3.2)$$

Thus, when a vector \mathbf{p} is presented, the weights of the winning neuron and its close neighbors move toward \mathbf{p} . Consequently, after many presentations, neighboring neurons learn vectors similar to each other[8]. Hence, the SOM network learns to categorize the input vectors it sees.

The SOM network used here contains N nodes ordered in a two-dimensional lattice structure. In these cases, each node has 2 or 4 neighboring nodes, respectively. Typically, a SOM has a life cycle of three phases: the learning phase, the training phase and the testing phase.

C. Unsupervised Learning

During the learning phase, the neuron with weights closest to the input data vector is declared as the winner. Then weights of all of the neurons in the neighborhood of the winning neuron are adjusted by an amount inversely proportional to the Euclidean distance. It clusters and classifies the data set based on the set of attributes used. The learning algorithm is summarized as follows[4]:

1. Initialization: Choose random values for the initial weight vectors $w_j(0)$, the weight vectors being different for $j = 1, 2, \dots, l$ where l is the total number of neurons.

$$w_i = [w_{i1}, w_{i2}, \dots, w_{il}]^T \in \mathbb{R}^n \quad (3.3)$$

2. Sampling: Draw a sample x from the input space with a certain probability.

$$x = [x_1, x_2, \dots, x_l]^T \in \mathbb{R}^n \quad (3.4)$$

3. Similarity Matching: Find the best matching (winning) neuron $i(x)$ at time t , $0 < t \leq n$ by using the minimum distance Euclidean criterion:

$$i(x) = \arg \min_j \|x(n) - w_j\|, \quad j = 1, 2, \dots, l \quad (3.5)$$

4. Updating: Adjust the synaptic weight vector of all neurons by using the update formula:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n) (x(n) - w_j(n)) \quad (3.6)$$

where $\eta(n)$ is the learning rate parameter, and $h_{j,i(x)}(n)$ is the neighborhood function centered around the winning neuron $i(x)$. Both $\eta(n)$ and $h_{j,i(x)}(n)$ are varied dynamically during learning for best results.

5. Continue with step 2 until no noticeable changes in the feature map are observed.

Training images are mapped into a lower dimension using the SOM network and the weight matrix of each image stored in the training database. During recognition trained images are reconstructed using weight matrices and recognition is through untrained test images using Euclidean distance as the similarity measure. Training and testing for our system was performed using the MATLAB Neural Network Toolbox.

D. Training

During the training phase, labeled DCT-vectors are presented to the SOM one at a time. For each node, the number of “wins” is recorded along with the label of the input sample. The weight vectors for the nodes are updated as described in the learning phase. By the end of this stage, each node of the

SOM has two recorded values: the total number of winning times for subject present in image database, and the total number of winning times for subject not present in image database[2].

E. Testing

During the testing phase, each input vector is compared with all nodes of the SOM, and the best match is found based on minimum Euclidean distance, as given in (3.5)[2]. The final output of the system based on its recognition, displays if the test image is “present” or “not present” in the image database.

IV. EXPERIMENTAL RESULTS

A. Image Database

A face image database was created for the purpose of benchmarking the face recognition system. The image database is divided into two subsets, for separate training and testing purposes. During SOM training, 25 images were used, containing five subjects and each subject having 5 images with different facial expressions. Fig. 7 shows the training and testing image database constructed.

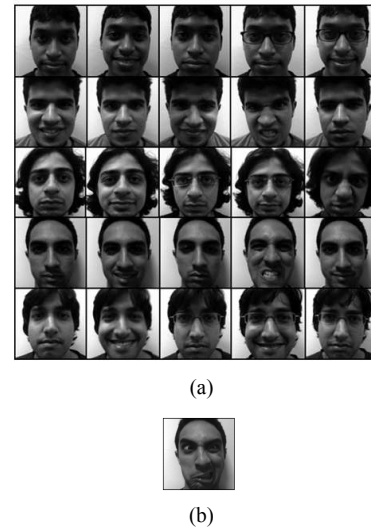


Fig. 7. Training and testing image database. (a) Image database for training. (b) Untrained image for testing

The face recognition system presented in this paper was developed, trained, and tested using MATLAB™ 7.2. The computer was a Windows XP machine with a 3.00 GHz Intel Pentium 4 processor and 1 GB of RAM.

B. Validation of Technique

The preprocessed grayscale images of size 8×8 pixels are reshaped in MATLAB to form a 64×1 array with 64 rows and 1 column for each image. This technique is performed on all 5 test images to form the input data for testing the recognition system. Similarly, the image database for training uses 25 images and forms a matrix of 64×25 with 64 rows and 25 columns. The input vectors defined for the SOM are distributed over a 2D-input space varying over $[0 \ 255]$, which represents

intensity levels of the grayscale pixels. These are used to train the SOM with dimensions [64 2], where 64 minimum and 64 maximum values of the pixel intensities are represented for each image sample. The resulting SOM created with these parameters is a single-layer feed forward SOM map with 128 weights and a competitive transfer function. The weight function of this network is the negative of the Euclidean distance[3]. This SOM network is used for all subsequent experiments. As many as 5 test images are used with the image database for performing the experiments. Training and testing sets were used without any overlapping. Fig. 8 shows the result of the face recognition system simulated in MATLAB using the image database and test input image shown in Fig. 7.

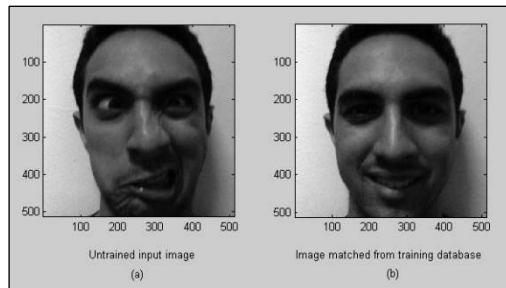


Fig. 8. Result of face recognition system. (a) Untrained input image for testing. (b) Best match image of subject found in training database

The result obtained from this simulation identifies that the subject in the input image Fig. 8(a) is “present” in the image database. The best match image displayed in Fig. 8(b) illustrates that subjects with different facial expressions in the image database can be easily identified. Euclidean distance for DCT-feature vectors for the untrained image database and SOM trained image database is shown in Fig. 9.

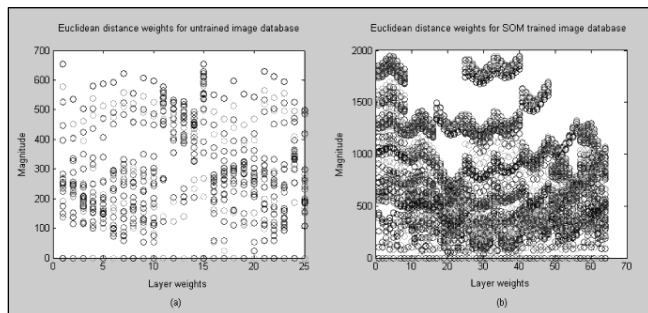


Fig. 9. Euclidean distance weights. (a) Untrained image database. (b) Image database trained with SOM network

As is clear from Fig. 9(a), the untrained DCT-feature vectors are represented by scattered clusters separated by large Euclidean distances. After SOM training, the 25-dimensional image database is transformed into a 64-dimensional map where the magnitude of the layer weights is increased, as shown in Fig 9(b). This transformation produces better classification by grouping similar clusters together, separated by smaller Euclidean distances[3]. A 3D-SOM network map

for this simulation displaying its neuron positions is shown in Fig 10(a). Weights vectors of the untrained image database in Fig. 10(b) and trained image database in Fig. 10(c) in comparison to the simulated 3D-SOM map in Fig. 10(a) are shown below.

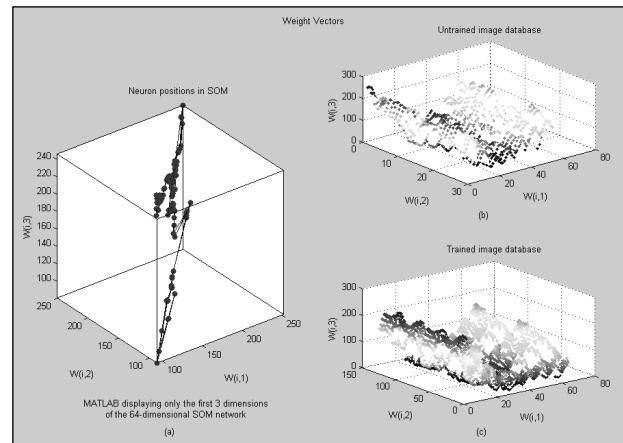


Fig. 10. Weight vectors of SOM. (a) Simulated 3D-SOM. (b) Untrained SOM. (c) Trained SOM

The next section presents results for 3 experiments in which different system parameters were altered. In the first two experiments the number of epochs used for training is 1000, and all experimental results obtained are the average of three consecutive simulations.

C. DCT Block Size

The first experiment studies the effect of DCT block size on the rate of recognition of the system with each DCT coefficient being used in the feature vector[2]. Table I shows that the best recognition rate obtained is for the case of 8×8 block sizes, and this block size was used in all subsequent experiments.

TABLE I
RECOGNITION RATES VS. DCT BLOCK SIZE
Maximum recognition rate is obtained at (8×8) block

	DCT block size					
	4x4	6x6	8x8	10x10	12x12	16x16
Recognition rate (%)	73.31	77.43	78.82	75.82	74.64	73.64

D. Reducing DCT-feature Vector Size

The second experiment is concerned with computational load, which comes from large sized DCT-feature vectors[2]. The aim of this experiment is to determine if a smaller feature vector could be constructed from a set of DCT coefficients, without significantly degrading system performance.

The current chosen DCT block size of 8×8 pixels uses only 8 out of 64 DCT coefficients for computation. Using statistical

analysis, by assessing the variance of each of the 64 dimensions of the feature space, it is possible to determine which of the coefficients contribute most to the final decision of the classifier. Variances were computed using[2]:

$$\text{var}(x_j) = \sum_{i=1}^k (X_i - \bar{X})^2 \quad (4.1)$$

where variable j is the DCT coefficient index, i is the sample index, and k is equal to the available number of samples.

Statistical variances of the 64 DCT-feature vectors for the training database and testing images indicated that there are 2 prominent areas where high-variance occurs. This suggests that these particular features perform prominent roles during classification[2]. Hence, we defined a reduced size feature space based on the 2 high-variance DCT coefficients and excluded other coefficients. The results obtained revealed that the new DCT-feature vectors consisted of 4 DCT coefficients only. Table II compares the performance of the system for the full size and reduced-size DCT-feature vectors. In spite of the dramatic reduction from 8 features to only 4, the recognition rates obtained are essentially the same. In addition to recognition rates, the table also shows the training time and memory usage.

This experiment demonstrates that good face recognition performance is possible, even with feature vectors that are dramatically reduced in size relative to the usual case for DCT-based analysis[2].

TABLE II
THE EFFECT OF REDUCING DCT-FEATURE VECTOR SIZE

DCT coefficients	Training Time (s)	Recognition rate	Memory consumption (bytes)
4	96.81	79.43%	2836740
8	163.05	78.82%	6589430

TABLE III
REDUCING PROCESSING TIME BY OPTIMIZING NUMBER OF EPOCHS FOR TRAINING

Number of epochs	Training time (s)	Recognition rate
700	49.97	73.03%
750	57.21	77.91%
800	63.94	78.54%
850	72.53	81.36%
900	85.40	80.51%
1000	96.81	79.43%
1050	108.94	75.98%

E. Reducing Processing Time based on Epochs

The third experiment is concerned with the processing time of the overall system. Processing time contributes mainly

towards the time required for training the SOM network. Training time depends upon the number of epochs used for training[3]. The aim of this experiment is to reduce training time, while maintaining the previously calculated recognition rate in experiment 2 for the reduced DCT-feature vectors. Table 3 shows that the best recognition rate achieved with the least amount of processing time is for the case of 850 training epochs. Recognition rate results obtained are the average of ten consecutive simulations.

V. CONCLUSION

This paper has presented a novel face recognition technique that uses features derived from DCT coefficients, along with a SOM-based classifier. The system was evaluated in MATLAB using an image database of 25 face images, containing five subjects and each subject having 5 images with different facial expressions. After training for approximately 850 epochs the system achieved a recognition rate of 81.36% for 10 consecutive trials. A reduced feature space, described for experiment 2 above, dramatically reduces the computational requirements of the method as compared with standard DCT-feature extraction methods. This makes our system well suited for low-cost, real-time hardware implementation. Commercial implementations of this technique do not currently exist. However, it is conceivable that a practical SOM-based face recognition system may be possible in the future.

REFERENCES

- [1] E. Hjelmås, and B. K. Low, "Face detection: A survey", *Computer Vision and Image Understanding*, Vol. 83, No. 3, Sept. 2001, pp. 236-274.
- [2] A. Abdallah, M. Abou El-Nasr, and A. Lynn Abbott, "A New Face Detection Technique using 2D DCT and Self Organizing Feature Map" in *Proc. of World Academy of Science, Engineering and Technology*, Vol. 21, May 2007, pp. 15-19.
- [3] J. Nagi, "Design of an Efficient High-speed Face Recognition System", Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, March 2007.
- [4] D. Kumar, C.S. Rai, and S. Kumar, "Face Recognition using Self-Organizing Map and Principal Component Analysis" in *Proc. on Neural Networks and Brain, ICNNB 2005*, Vol. 3, Oct 2005, pp. 1469-1473.
- [5] E. Ifeachor and B. Jervis, *Digital Signal Processing: A Practical Approach*. Prentice Hall, 2001.
- [6] L. Ma, Y. Xiao, and K. Khorasani, "A new facial expression recognition technique using 2D DCT and k-means algorithm", in *Proc. International Conference on Image Processing*, Oct. 2004, pp. 1269-1272.
- [7] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks", *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol. 34, No. 3, June 2004, pp. 1588-1595.
- [8] Y. Zi Lu and Z. You Wei, "Facial Expression Recognition Based on Wavelet Transform and MLP Neural Network", in *Proc. 7th International Conference on Signal Processing, ICSP 2004*, Vol. 2, Aug 2004, pp. 1340-1343.