

Military Technical College
Kobry Elkobbah,
Cairo, Egypt



2nd International Conference on
Electrical Engineering
ICEENG 99

A NEURAL NETWORK APPROACH FOR BLOCK CODING IMAGE COMPRESSION

*M. SHAARAWY IBRAHIM **

ABSTRACT

This paper presents a scheme for image compression using block coding by vector quantization technique. This scheme achieves promising results in compression ratio and image quality. Although the encoding time represents a big problem in such solutions, it is improved using a self-organizing feature map (SOFM) neural network. Feature extraction reduces the dimensionality of the problem and enables the neural network to be trained on an image separate from that for testing. Although the time complexity has been reduced, the image quality is also affected by a slight value, which can be accepted in many situations.

KEYWORDS: Block Coding, Vector Quantization, Self-Organizing Feature Map, Moment Invariant, Feature Extraction.

* Associate Professor, Dpt. of Computer & OR, Military Technical College, Cairo, Egypt.

INTRODUCTION

Image compression methods are divided into two categories, Lossless and Lossy compression methods. The compression method is called Lossless when the original image can be exactly reconstructed from the compressed version without any loss in the image information while the compression method is called lossy when an approximated image is reconstructed from the decompressed image file [1]. Unfortunately, compression with lossless methods seldom exceeds 2-3:1 and there is a little chance to expect new algorithms to offer more significant improvement. Lossy methods, in contrast, can achieve much higher compression ratios but at the cost of introducing artifacts and distortions. The challenge in image compression research is to develop some compression methods that combine the three desirable goals; deep compression ratio, low image distortion, and high computation speed. Many approaches have been developed to achieve one or two of the mentioned goals, and a few achievements for the three goals. For example, block coding image compression techniques, such as vector quantization (VQ) or fractal image compression techniques achieve the first and second goals, but the time encoding complexity represents a main problem [2,3].

This paper presents a scheme for improving encoding time for image compression by using a self-organizing feature map (SOFM) neural network for vector quantization technique combined with some feature extraction for range-domain classification. Feature extraction reduces the dimensionality of the problem and enables the neural network to be trained on an image separate from that for testing. The network organizes itself according to the distribution of the image features observed during training [4]. In section 2, the Kohonen Self-Organizing Feature Map concept is introduced. Section 3 explains briefly the main concept of vector quantization and the use of SOFM in VQ coding. The different types of feature extraction, which can be used for domain and range classification are introduced in section 4. System performance is introduced in section 5. Finally a conclusion is presented.

2- KOHONEN'S SELF-ORGANIZING FEATURE MAP:

Kohonen uses spatial mapping to model complex data structures internally. This concept enables to perform data compression on the vectors stored in the network. This also allows the network to store data in such a way that the spatial or topological relationships in the training data are maintained and represented in a meaningful way.

SOFM has been used in block coding methods such as vector quantization (VQ) with application to speech and images. In block coding algorithms the image is partitioned into blocks of pixels of equal size: b_{ij} , $(i,j) \in I$ with I the set of block indices, each block being individually coded. Usual block sizes are 4×4 , 8×8 or 16×16 for images of 512×512 pixels. Each block is an element (vector) in a high dimensional space V (a 8×8 block in R^{64}). The feature mapping property of the SOFM is used to map the clusters from V , which represents the space spanned by the partition blocks onto the reduced dimension array of formal neurons. The similarity or pattern

closeness between two blocks is reflected in the position (distance in the neuron index) of the neurons associated with the blocks. The vector space spanned by the blocks is quantized due to the fewer number of neurons in the network, which represent an image or a set of images. The VQ block coding process is divided into two parts; first, the building of the codebook (quantization) where each block in the codebook is assigned an index. Second, the code matching of each image partition block with an element from the codebook followed by the assignment of the index of the coded block. In a neural network (NN) frame work, a code book is built by training the SOFM with a set of blocks from the test image D. At the end of the training, each neuron will store its weights which represent an approximation of the blocks of the training set. A block is applied to the network input nodes and the neuron with the highest index (position in the network) will represent the code of the blocks.

2.1-The Learning Algorithm of SOFM

Kohonen's self-organizing feature (SOFM) map may be used to find clusters in a set of data vectors stemming from a stochastic source $x \in R^n$. The network training is made through unsupervised learning. The weight vectors are initialized to random values. An input feature vector is then presented to the network and weight vector closest to the input vector is selected as the winner vector. The algorithm starts by assigning to every node i of a regular array an initial cluster center $m_i(0) \in R^n$. At step t , a vector $x(t)$ is taken from a training set $\{x(t) \in R^n, t = 0, \dots, T\}$ and compared to all clusters $m_i(t)$. Let $c = \arg(\min_i \|x(t) - m_i(t)\|)$ Then $m_c(t)$ and all cluster centers associated to nodes in the neighborhood of c are moved to the input vector $x(t)$ by the following rule

$$m_i(t+1) = m_i(t) + \alpha(t) \tag{1}$$

SOFM training belongs to the class of competitive learning algorithms. The network consists of the input layer and the output competitive layer arranged as a rectangular array of formal neurons as shown in figure 1. There is only one layer of neurons and all inputs are connected to all nodes. A neuron N_{mn} (mn is the position of the neuron in the SOFM) is a simple processor having a rectangular array of weights w_{mn} the size of a partition block b_{ij} from the image D. The input b_{ij} is applied in parallel to all the neurons each neuron has a output:

$$u_{mn} = d(w_{mn}, b_{ij}) \tag{2}$$

Where $d(., .)$ is a similarity measure in V .

Network training is made through unsupervised learning. The training sequence of pixel blocks b_{ij} is selected from a test image D or several images. The network of the weight vectors is initialized to random values. An input vector is then presented to the network and the weight vector closest to the input vector is selected. The learning algorithm can be described as follows:

1. Initialize the weights w_{mn} of each neuron N_{mn} with random values or use a predefined map obtained from previous training.
2. A block b_{ij} is presented to the network and each output neuron computes the distortion u_{mn} .
3. Select the winning neuron N_{max} , which stores the closest pattern to the input b_{ij} .
4. Update the weights of the selected neuron and its neighbors:

$$w_{mn}^t = w_{mn}^{t-1} + \eta(b_{ij} - w_{mn}^{t-1}) \quad (3)$$

where η is the adaptation step size such that $0 < \eta < 1$.

5. Repeat steps 2) to 4) for all training vectors.

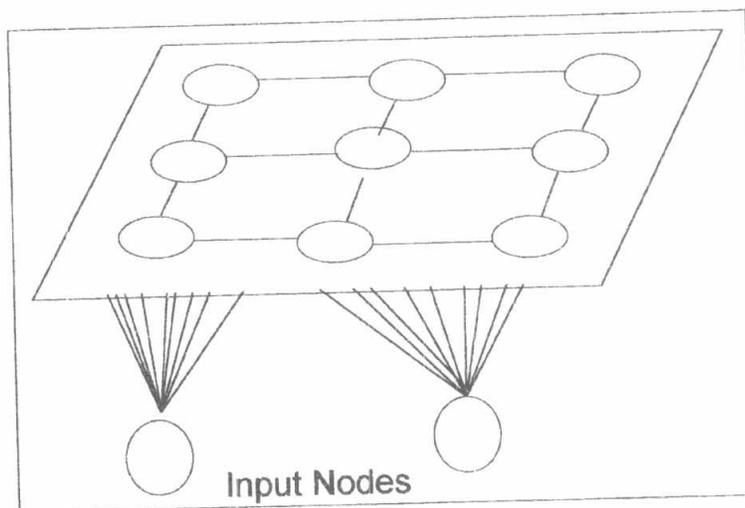


Figure 1: Kohonen's SOFM neural network

The size of the neighborhood is reduced during the course of the training epochs. The parameter η is the epoch step size, which is inversely proportional to the neighbor step size [5]. Once the network has been trained, the domain cells for the given images are classified by assigning them to the weight vector to which they are closest in feature space. When a range cell feature vector is presented to the network, it is similarly assigned to a network vector. The range cell is then compared to those domain cells assigned to this weight vector, as well as to those domains assigned to the neighboring weight vectors on the network lattice.

An advantage of the self-organizing network approach for classification is this concept of neighboring image classes, which is provided by the network lattice. Another advantage is that this clustering of image types occurs without the need to predefine image classes.

3- VECTOR QUANTIZATION (VQ)

In standard Vector Quantization compression schemes, each range block is paired to an identically sized domain block stored in a separate file. This file is called the *codebook* and consists of an ordered list of representative domain blocks. The challenge is to design a code book with a good variety of domain blocks which contains some varieties that represents smooth, sloping, edges, stripes and fine details portions of the image while keeping the size of the code book small.

3.1- SOFM in VQ Coding

In VQ coding algorithm, the picture is divided into non-overlapping blocks equal to the number of the blocks in the codebook. After finishing the training process, each block b_{ij} from the partition is replaced by the corresponding neuron N_{kl} from the SOFM such that:

$$kl = \arg(\min_{mn} d(w_{mn}, b_{ij})) \quad (4)$$

Thus, the index kl is the code for the block b_{ij} . In telecommunication applications, the index of N_{kl} for each block are transmitted and the image is reconstructed at the receiver by replacing these indices by the corresponding block/vector from the code-book presented at the receiver. One major advantage of using the SOFM over the arbitrary indexing of the VQ code-book, is that, if any information in the index has been corrupted or partially lost then the remaining information can take over, pointing to a neighboring neuron or to an area of the SOFM such that the reconstructed image will have a minimal degradation [6].

4- SOME TYPES OF FEATURE EXTRACTION

One way to improve the basic image coding process is to extract a number features which characterize the domain and range subimages. The range-domain matching is based on these features rather than the individual pixels, thus reducing the complexity of the problem and providing a significant speedup in the encoding process. In other words, feature extraction has two advantages. First, the reduction in dimensionality of the problem, which reduces the computations, needed during the search matching. Second, it is possible to train the self-organizing network on one image and apply the resulting network for classification on a different image due to the feature independence of the structure of particular domain. In this section some types of feature extractions are presented and then they are combined with SOFM neural network in the next section.

4.1- Moment Invariant

They are a set of moments used for identification and scene matching [7,8]. This set of moments has been shown to be invariant to translation, rotation and scale change and they revised and modified by Sidhartha [9].

4.2- Eign Vector

The second feature to extract from the domain blocks is the eign values of each block. The domain blocks are discriminated in the domain pool according to their eign vector position and magnitude. To match each range with a domain block (in the domain pool), the range eign vector is calculated and compared to the eign vectors of the domains. Then a match is made with the minimum distance measure, which represent the best match between the range and the domains [10].

4.3- A Set of Feature Vectors

Six features are used by [4] to measure image texture and contrast distribution. These six features are:

- (1) The average of the block.
- (2) The standard deviation (σ variance).
- (3) Skewness, which sums the cubes of the differences between pixel values and the cell mean, normalized by the cube of σ .
- (4) Neighbor contrast, which measures the average difference between adjacent pixel values.
- (5) Horizontal gradient, which measures the changes in the horizontal pixel values across the cell.
- (6) Vertical gradient, which measures the changes in the vertical pixel values from top to bottom.

5- SYSTEM PERFORMANCE

The set of projected codebook blocks are partitioned into a finite number of disjoint subsets (clusters) defined by representatives (cluster centers) such that vectors in the same cluster are closer to each other than vectors in different clusters. The quality of the clustering depends on the criterion function used for distance measured. In this section, the standard image of Lena 128×128 is used for training the SOFM. The network training time is not a part of the overall encoding time. Once the feature map (SOFM) has been trained, it can cluster any new image applied to it even it is of different size or has different nature as shown below.

The image quality is measured by the peak signal-to-noise ratio (PSNR). PSNR for two image object μ and ν is given by:

$$PSNR(\mu, \nu) = 10 \log_{10} \left[\frac{(dr - 1)^2}{RMS(\mu, \nu)^2} \right] \quad (5)$$

Where dr is the dynamic range of the image object. For an eight bit per pixel image object, the dynamic range is 256 pixel values. RMS is the root mean square error.

In figure 2, the 512×512 Lena image is reconstructed using the weights of SOFM which used 128×128 lina image as a training image and the codebook size has 70 clustering elements, achieving Peak Signal to Noise Ratio (PSNR) of 27.05 dB. It is

clear that, the same image has been reconstructed even it has different resolution than the image used for training. Figure 3 shows the number of classes associated by every cluster in the codebook representing the image. Reconstructing images of different resolutions is not the only advantage of SOFM, but another important improvement can be achieved by the use of Kohonen map where the reconstruction of images is applicable even if they are of different nature from that used in training the map. Figure 4 shows the reconstruction of Boat 512×512 image from the training image Lenna 128×128 and figure 5 shows the number of classes associated by every cluster in the codebook representing the image. Note that, these two images are different in nature and size.



Figure 2: Lena 512×512 reconstructed from the weights of code book of 70 block and PSNR = 27.05 dB.

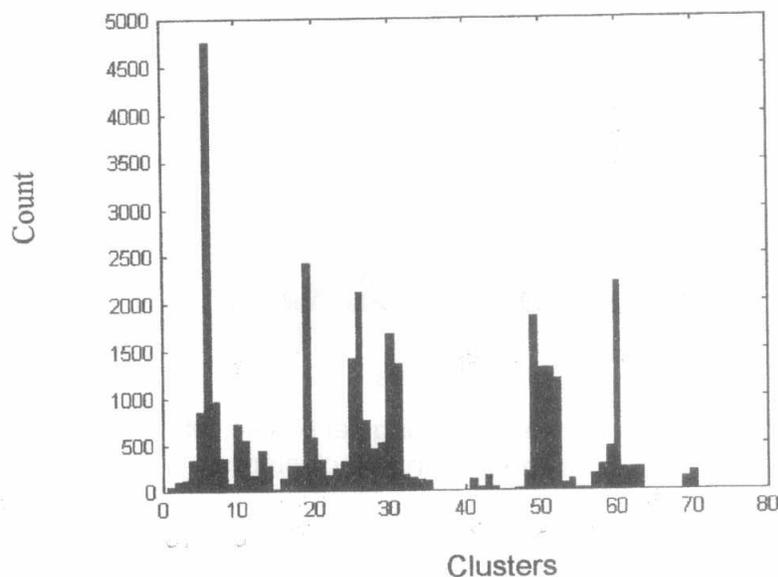


Figure 3: Count of the clusters of original blocks, when coding Lena image using 70 clusters

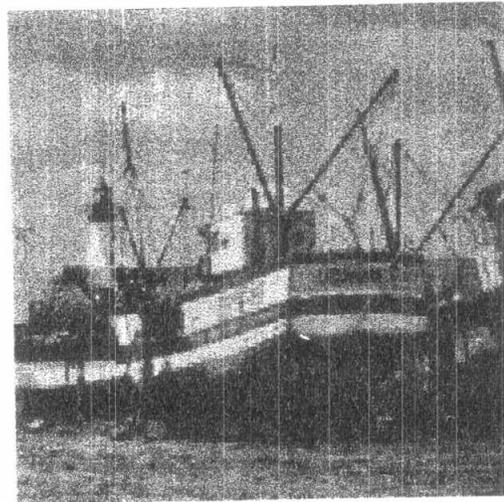


Figure 4: Boat 512×512 reconstructed from the training image Lena 128×128 with PSNR = 24.64 dB.

From figure 3 and figure 5 there exists some clusters which gain no classes of range blocks. This is because the codebook-generated during training by Lena image-contains vectors, which are useful for generating other images. Hence, during image reconstruction vectors are matched to the nearest match. This represents the main reason for image quality reduction by a value from 3 to 5 dB.

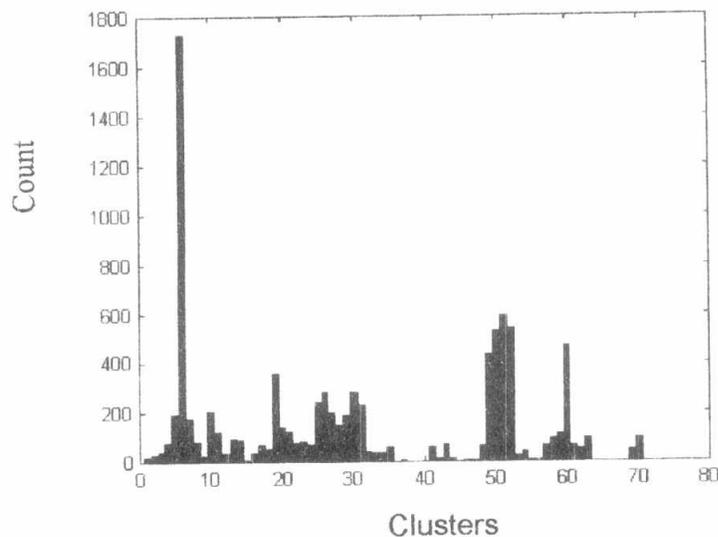


Figure 5: Count of the clusters of original blocks, when coding the Boat image using 70 clusters



Figure 6: Lynda 512×512 reconstructed from Lenna 128×128 with code book of 70 blocks and PSNR = 21.13 dB.

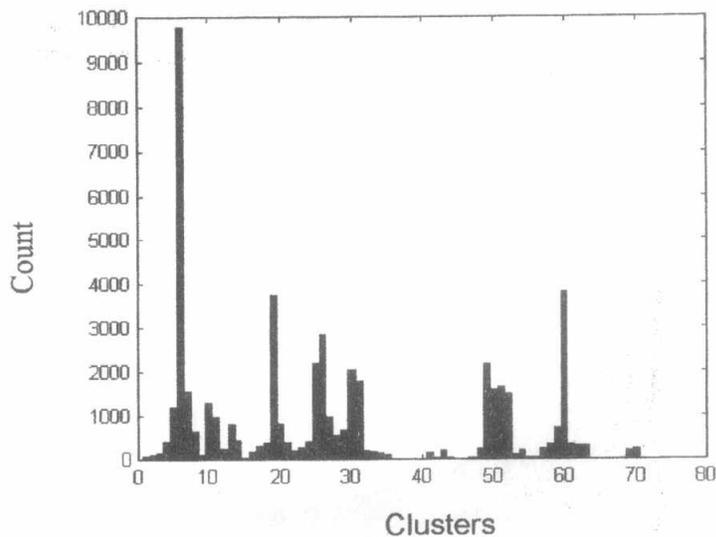


Figure 7: Count of the clusters of original blocks, when coding Lynda image using 70 clusters. The image contains 512×512 pixels and the blocks consists of 4×4 pixels.

The map, which also has different size checks another standard image of Lynda, Figure 6, than the image used for training.

The distribution of range classes in figure 7 has less normal nature than that of figure 3 and figure 5 and this is the reason that figure 6 of Lynda has less quality (PSNR=21.13 dB) than the two previous images.

During training process, the neighbors start with the number of the codebook-1 then decreases rapidly in the next epoch then stays at value one to the end of the training process as shown in figure 8. On the other hand, the error of the learning rate starts

with high value at the beginning of the training process then decreases to a very low value at the end of the process as shown in figure 9.

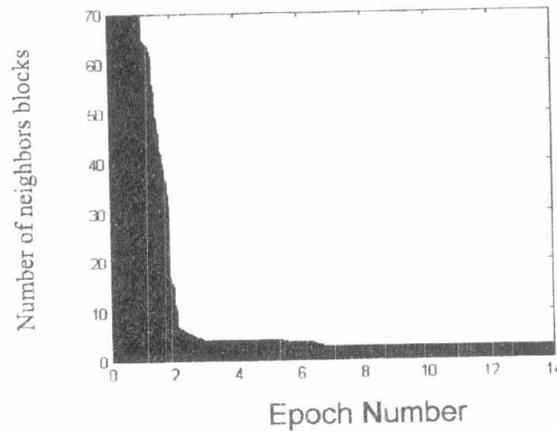


Figure 8: The Number of neighbors blocks decreases during training time.

The codebook of size 70 blocks and block size 4×4 which is made up from Lena 128×128 can be used to reconstruct any other images with any resolution with block size 4×4 pixels.

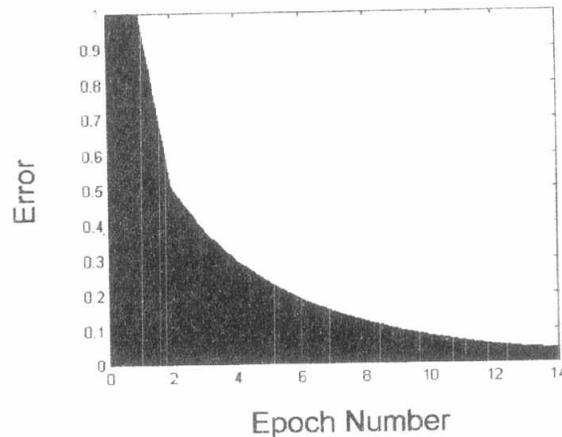


Figure 9: The error during the training time.

5.1- Time Performance:

The time complexity of the encoding is one of the major drawbacks of block coding image compression. The time consuming part of the encoding step is the search for an appropriate domain for each range. The number of possible domains that theoretically may serve as candidates is very large. Each range has to be compared sequentially to each domain, which are overlapping blocks through the entire image with eight different transformation symmetries. Such a method needs:

$$N_D \times N_R \times 8 \text{ Operations}$$

where:

N_D is the number of domain blocks,
 N_R is the number of range blocks,
 and the number of different symmetries is eight symmetries in different directions of flipping and rotation.

Thus, the order of computations needed is of order $O(N^2)$.

In case of using the SOFM, the situation is different. The training of the map has been made by using the Neural Net ToolBox of the Matlab package (ver. 4) on a Pentium PC (233 MHZ). It has taken around 4:32 hours to train the map using Lena 128x128 as an input training image for the net. If the training time for the map is excluded because it is just made once for the map, then the computational time needed depends on two things:

The number of ranges N_R and the codebook size C_B . Thus the order of the search needed is of order $O(N)$. Practically, the standard Lena image of size 512 x 512 has taken 342 seconds (\approx 5.7 minutes) to find the range-codebook matching.

Although the SOFM has reduced the time complexity from the order $O(N^2)$ to the order $O(N)$ but this has an effect on the image quality and the Compression Ratio (CR).

5.2- Image Quality

Although the time complexity has been reduced to the order N , the image quality is also reduced by a value between 3 to 5 dB, which can be accepted in such situations. Table 1 represents applying different methods on lenna 128 x 128 as training image to the SOFM. In this situation vector quantization achieves the best PSNR, then feature set, Eign vector and last moment invariant. This is because vector quantization codebook may have more information representing the image than the others.

Table 1: Comparison between different methods of training the SOFM and the output image quality for Lena 512 x 512 image.

	Grayscale (VQ)	Feature Set	Eign Vector	Moments
PSNR dB	27.05	26.12	23.48	20.72

The output file of the map representing the weights of the net (codebook of size 70 clusters each of size 4x4 pixels) is saved by the Matlab package on 12,822 bytes \approx 12.5 KB. Thus, for example, if an image of size 512 x 512 (256 KB) is encoded then the CR of such an image is given by

$$CR = 256KB(\text{original size})/12.5KB(\text{output file}) = 20.44: 1$$

For images of higher resolutions, the compression ratio could be better.

CONCLUSION

Image compression techniques have to compromise between the three desirable goals; deep compression ratio, high image quality and reduced computation time. Block coding techniques are expected to achieve this compromise by the help of some new tools such as neural networks. The neural net approach reduces the time complexity search of the proposed algorithm to the order of n instead of order n^2 . One of the advantages of such new tools is that, any image file can make training of SOFM for the range-domain block matching, also the image reconstruction can be achieved for any resolution. Also, SOFM can be used for image decoding as well as image encoding to improve time performance. The cluster number of SOFM can be optimized to achieve higher compression ratio without affecting the image quality. Finally vector quantization achieves better image quality than other techniques used for feature extraction. This is because vector quantization codebook may have more information representing the image than the others.

REFERENCES

- [1] John Kominek. *"Understanding Fractal Image Compression"*. Department of computer Science, University of Waterloo, Canada, 1996.
- [2] Y. Fisher, D. Rogovin, T. P. Shen. *"Fractal (Self-VQ) Encoding of Video Sequences"*. Proceedings of the SPIE, Visual Communication and Image Processing '94 (VCIP'94), 1994. Found on WWW site [<http://legendre.ucsd.edu/Research/Fisher/fractal.html>].
- [3] Behnam Bani-Eqbal. *"Speeding Up Fractal Image Compression"*. Department of Computer Science., University of Manchester, UK, 22 September 1994. Paper found on ftp site 'ftp. informatik.uni-freiburg.de' in 'papers/fractals' email: behnam@cs.man.ac.uk.
- [4] Stephen Welstead, "Self-Organizing Neural Network Domain Classification for Fractal Image Coding", Proc. of the IASTED International Conference Artificial Intelligence And Soft Computing, July 27-31, 1997, Banff, Canada pp. 248-251, 1997.
- [5] R. Beale , T. Jackson. *"Neural Computing: An Introduction"*. Adam Hilger. Bristol, Philadelphia and New York, 1990.
- [6] A. Bogdan and H.E. Meadows, *"Kohonen neural network for image coding based on iteration transformation theory"*, SPIE's 92 International Symposium, San Diego, CA., 1992.
- [7] M. K. Hu. *"Visual Pattern Recognition By Moment Invariants"*. IRE Transactions on Information Theory, IT-8:275-282, 1962.
- [8] R. C. Gonzalez, R. E. Woods. *"Digital Image Processing"*. Addison-Wesley, 1992.
- [9] Sidhartha Maitra. *"Moments Invariants"*. Proceeding of the IEEE, Vol. 67, No. 4, April 1979.
- [10] M. Shaarawy, H. Ismail and K. Hassanain, "A Face Verification System Based Upon Eigenface Approach", Proceeding of 22nd ICC & IE Conference, Cairo - Egypt, 20-22 Dec, 1997.