A Complexity-Based Approach in Image Compression using Neural Networks

Hadi Veisi, Mansour Jamzad

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

***Abstract*—**In this paper we present an adaptive method for image compression that is based on complexity level of the image. The basic compressor/de-compressor structure of this method is a multi-layer perceptron artificial neural network. In adaptive approach different Back-Propagation artificial neural networks are used as compressor and de-compressor and this is done by dividing the image into blocks, computing the complexity of each block and then selecting one network for each block according to its complexity value. Three complexity measure methods, called Entropy, Activity and Pattern-based are used to determine the level of complexity in image blocks and their ability in complexity estimation are evaluated and compared. In training and evaluation, each image block is assigned to a network based on its complexity value. Best-SNR is another alternative in selecting compressor network for image blocks in evolution phase which chooses one of the trained networks such that results best SNR in compressing the input image block. In our evaluations, best results are obtained when overlapping the blocks is allowed and choosing the networks in compressor is based on the Best-SNR. In this case, the results demonstrate superiority of this method comparing with previous similar works and JPEG standard coding.

***Keywords*—**Adaptive image compression, Image complexity, Multi-layer perceptron neural network, JPEG Standard, PSNR.

I. Introduction

IMAGE data compression continues to be an important subject in many areas such as communication, data storage, and computation. The existing traditional techniques mainly are based on reducing redundancies in coding, inter-pixel and psycho visual representation [1]. In addition, new soft computing technologies such as neural networks are being developed for image compression. Parallelism, learning capabilities, noise suppression, transform extraction, and optimized approximations are some main reasons that encourage researchers to use artificial neural networks as an image compression approach. Although there are no significant work on neural networks that can take over the existing technology but there are some admissible attempts. Research activities on neural networks for image compression do exist in many types of networks such as - Multi-Layer Perceptron (MLP) [2-13], Hopfield [14], Self-Organizing Map (SOM), Learning Vector Quantization (LVQ) [15,16], and Principal Component Analysis (PCA) [17]. Among these methods, the MLP network which usually uses back-propagation training algorithm provides simple and effective structures. It has been more considered in comparison with other artificial neural network (ANN) structures.

The compression of images by Back-Propagation Neural Networks (BPNN) is investigated by many researchers. One of the first tries in using this approach was done in [3], in which the authors proposed a three layer BPNN for compressing images. In their method original image is divided into blocks and fed to input neurons, compressed blocks are found at the output of the hidden layer and the de-compressed blocks are restored in the neurons of output layer. This implementation was done on the NCUBE parallel computer and the simulation results showed that this network could achieve a poor image quality even for trained images in 4:1 compression ratio [3]. As in [11] pointed out, none of the results in using single network are so good as the result that could be achieved by taking average of image blocks and using their values as the indicator of blocks!. Because of these poor results achieved by using one simple BPNN, several authors tried to improve the performance of this neural network-based compression technique. One of these efforts was hierarchical neural networks [13] which extended BPNN by adding two more hidden layers to it. This extension will exploit the correlation between blocks in an image in addition to the correlation between pixels among a block. This method had some improvements in SNR of reconstructed image, but this improvement is not so considerable.

Adaptive methods use another approach to compress/de-compress (CODEC) the image blocks. In this approach various networks are used for compress/decompress different image blocks regarding to the complexity of blocks. It provides best results in compression with neural networks. In [6,7] it is suggested to cluster image blocks into some classes based on a complexity measure called *activity*. They have used four BPNNs with different compression rates for each class. This yielded significant improvement over basic BPNN. Another adaptive approach which proposed the use of complexity measure with block orientation by six BPNNs has given better visual quality [11]. An extension of this approach is given in [8] in which blocks are classified into nine predefined orientations for reducing edge degradation. In this method different networks were used for compressing the blocks in each class. The BPNNs were used for compressing image blocks, after that each pixel in a block was subtracted from the mean value of the block. This method gives some

Manuscript received February 05, 2007.

Hadi Veisi is with Computer Engineering Department of Sharif University of Technology, Tehran, Iran, (phone: +98-021-66003710; fax: +98-021-66003710; e-mail: veisi@ce.sharif.edu).

Mansour Jamzad is with Computer Engineering Department of Sharif University of Technology, Tehran, Iran, (e-mail: jamzad@sharif.edu)

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

good results although accommodate extra overheads in transmitting the average values.

In this paper we have used the basic neural network-based algorithm for compressing images. Then an adaptive approach for compression is presented. We have proposed methods for computing the complexity of image blocks that are based on the concept of Entropy, Activity, and Pattern trajectory in blocks. This adaptive approach utilizes various BPNNs with different compression ratios that are used to compress/de-compress image blocks depending on the level of complexity in the block. In practice, we have used the complexity criterion to select the appropriate network for compressing incoming image block. Also Best-SNR method is used to select the network that gives the best SNR for that image block. In addition, overlapping of image blocks is used in order to eliminate the chess-board effect in de-compressed image. Our experimental results showed that composition of overlapping blocks and choosing the network with Best-SNR yield improvements in PSNR and visual quality of reconstructed image compared to standard and conventional JPEG coding.

This paper is organized as follows. In section II we discuss multi-layer perceptron neural network and its adaptive approach that is directly developed for image compression. Section III describes the complexity measurement methods used in this paper. In section IV, the experimental results of our implementations are discussed and finally in section V we conclude this research and give a summary on it.

II. Multi-Layer Neural Networks for Image Compression

Multi-Layer neural networks with back-propagation algorithm can directly be applied to image compression. The simplest neural network structure for this purpose is illustrated in Fig. 1. This network has three layers, input, hidden and output layer. Both the input and output layers are fully connected to the hidden layer and have the same number of neurons, N. Compression can be achieved by allowing the value of the number of neurons at the hidden layer, K, to be less than that of neurons at both input and output layers ($K \le N$). As in most compression methods, the input image is divided apart into blocks, for example with 8×8, 4×4 or 16×16 pixels. These block sizes determine the number of neurons in the input/output layers which convert to a column vector and fed to the input layer of network; one neuron per pixel. With this basic MLP neural network, compression is conducted in training and application phases as follow.

A. Training

Like all other training processes, in this phase a set of image samples are selected to train the network via the back-propagation learning rule. For compression purpose the target pattern in the output layer neurons of the network will be same as the input pattern. The compression is represented by the hidden layer which is equivalent to compress the input into a narrow channel. Training samples of blocks are converted into vectors and then normalized from their gray-level range into [0, 1]. In accordance with the structure of neural network shown in Figure1, the operation for adjusting weights for compressing and de-compressing can be described as the following equations.

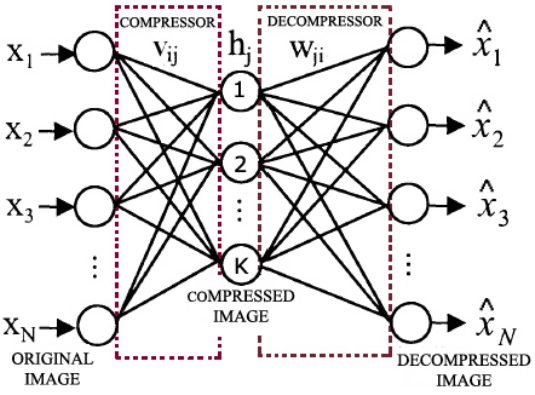


Fig. 1 Basic image compression structure using neural network

$$H_j^{in} = \sum_{i=1}^{N} V_{ij} X_i \ , \quad h_j = f(H_j^{in}); \ 1 \le j \le K \tag{1}$$

$$\hat{X}_i^{in} = \sum_{j=1}^{K} W_{ji} h_j \ , \quad \hat{X}_i = g(\hat{X}_i^{in}) \ ; \ 1 \le i \le N \tag{2}$$

In the above equations, f and g are the activation functions which can be linear or nonlinear. V_{ij} and W_{ji} represent the weights of compressor and de-compressor, respectively. The extracted $N \times K$ transform matrix in compressor and $K \times N$ in de-compressor of linear neural network are in direction of PCA transform. This transform provides optimum solution for linear narrow channel type of image compression and minimizes the mean square error between original and reconstructed image. In addition, it maps input samples into a new space where all samples in the new space are de-correlated; this fact led better compression. But unfortunately this is a data-dependent transform and it can only provide good compression for trained images. Using linear and non-linear activation functions in this network results linear and non-linear PCA respectively.

The training process of the neural network structure in Fig. 1 is iterative and is stopped when the weights converge to their true values. In real applications the training is stopped when the error of equation (3) reaches to a threshold named ε or maximum number of iterations limits the iterative process.

$$Err = \frac{1}{2} \sum_{k=1}^{N} \left(X_k - \hat{X}_k \right)^2 \tag{3}$$

B. Application

When training is completed and the coupling weights are adjusted, the test image is fed into the network and compressed image is obtained in the outputs of hidden layer.

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

These outputs must be quantized to the desired number of bits. If the same number of bits is used to represent input and hidden neurons, then the Compression Ratio (CR) will be the ratio of number of input to hidden neurons. For example, to compress an image block of 8×8, 64 input and output neurons are required. In this case, if the number of hidden neurons are 16 (i.e. block image of size 4×4), the compression ratio would be 64:16=4:1. But for the same network, if 32 bits floating point is used for coding the compressed image, then the compression ration will be 1:1, which indicates no compression has occurred. In general, the compression ratio of the basic network illustrated in the Fig. 1 for an image with n blocks is computed as Eq. (4).

$$CR = \frac{nNB_I}{nKB_H} = \frac{NB_I}{KB_H} \quad (4)$$

Where B_I and B_H are the number of bits needed to code the output of input and hidden layers, respectively. In this equation N and K are the number of neurons in the input and hidden layers, respectively.

In de-compressor, the compressed image is converted to a version similar to original image by applying the hidden to output layer de-compression weights on outputs of hidden layer. The outputs of output neurons must be scaled back to the original grayscale range, i.e. [0~255] for 8 bit pixels.

C. Adaptive approach

As mentioned in the previous section, the basic structure of neural network for image compression provides an approximation of PCA transform. This structure tries to de-correlate the input samples of pixels; this process is a major issue in data compression. But because of dependability of this transform to trained data, it is not used in many real applications. This is the main reason that PCA is replaced with its nearest approximate, the data-independent Discrete Cosine Transform (DCT) transform in real applications. Due to this limitation of the basic neural network structure for compression, the results obtained from this network show that in it is too weak to be used.

One method for improving the performance of this simple structure is the adaptive approach which uses different networks to compress blocks of the image [2,5-11]. Doing this, at first, image blocks are divided into several classes according to their complexity. Then image blocks of each class are used to train a network in the way the compression ratio of the network is related to the complexity of this class. All of the networks have identical structure, but they have different number of neurons in hidden layers, which will result in different compression ratios.

Considering the network of Fig. 1 as the basic structure, we can present the adaptive method as in Fig. 2. As it is shown in this figure, to train the networks the amount of information available in each block is estimated by means of a value according to a complexity measure criterion like average of the gray-levels in image block or some other methods. Then

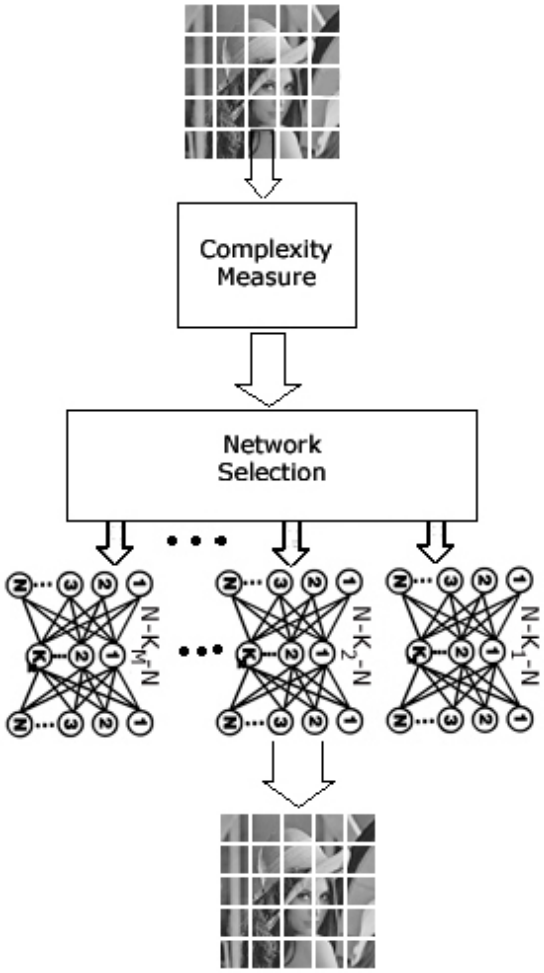


Fig. 2 Neural network-based adaptive structure for image compression

according to this complexity value, one of the available networks is selected. Each network is trained using its corresponding train data by Back-propagation algorithm.

To identify for de-compressor which network is used in compressor stage to compress the image block, a code is assigned to each trained network. This code should be transmitted or be saved along the compressed image. It is clear that the number of networks and consequently the number of bits needed to present this code will affect the compression ratio. The lower number of bits is preferred from the overhead view of point but on the other hand the, lower number of networks reduces the adaptively ability of the algorithm. In de-compressor this saved or transmitted code along with the compressed image is extracted and therefore, the corresponding network (i.e. the same network used in compression stage) can be selected for de-compression.

In adaptive approach, we assume to have M different networks with k_1 - k_M neurons in hidden layer. In this case for an image with n blocks each having N pixels, the compression ratio is as equation (5) that is obtained by modifying equation (4).

$$CR_a = \frac{n\,N\,B_I}{\sum_{i=1}^{n} B_H K_j^i + q} = \frac{NB_I}{(\frac{B_H}{n}\sum_{i=1}^{n} K_j^i) + \frac{q}{n}}; 1 \le j \le M \qquad (5)$$

In the above equation, K_j^i is the number of neurons in the hidden layer of selected network for i^{th} block image and $1 \le j \le M$. q is the number of bits that are needed to code the network number. In fact q is equal to the smallest positive integer such that $2^q \ge M$.

III. Complexity Measurement Methods

In the following three methods are presented for calculating the detail level of image blocks to incorporate in adaptive BPNN algorithm. Depending on the values of detail level, image blocks are classified into several classes. One network is assigned to each class and the compression ratio of that class is related to its complexity. In fact the complexity measurement criteria should reveal the amount of information in an image block. Also, it should be able to discriminate the image blocks according to neural networks-based compression. The complexity measure criterion is an important factor for this approach and it affects the compression performance, significantly. Here, we have used three different criteria, Entropy, Pattern-based and Activity.

A. Complexity based on Entropy

It is known that *Entropy* is a meaningful criterion to measure the amount of information in a set of symbols like an image. The entropy of an image block with N different gray-levels is calculated as (6). Where $P(x_i)$ is the probability of occurrence of gray-level x_i in this block.

$$Entropy = -\sum_{i=1}^{N} P(x_i)\log P(x_i) \qquad (6)$$

An image block that has a higher Entropy value contains more information. It means that, to prevent more loss of data, that block should go through less compression. Also, a block with lower Entropy value should be compressed by a network which provides higher compression ratio.

B. Complexity based on Activity

This measurement method is defined to cover the subjective idea of activity in an image block. For an image block with N pixels (i.e. in size $\sqrt{N}\times\sqrt{N}$) the *Activity* is defined as equation (7).

$$Activity = \sum_{i=2}^{\sqrt{N}}\sum_{j=2}^{\sqrt{N}}\left[\sum_{m=-1}^{1}\sum_{n=-1}^{1}(x_{i,j} - x_{i-m,j-n})^2\right] \qquad (7)$$
$$i, j = even;\quad (m,n) \ne (0,0)$$

In this measurement, low activity classes require networks with high compression rate and high activity classes need to maintain more data which means that they should use networks with larger number of hidden neurons and lower compression rate.

C. Pattern-based complexity

Although Activity is a good subjective criterion in complexity approximation and Entropy is a semantic measure for calculating the amount of information in a block of data, but in our usage of these methods for learning purpose, we faced some difficulties. It is clear that the Entropy values of all image blocks in Fig. 3-(5) to Fig. 3-(8) are equal. This is true for the Activity values, too. This is because these blocks have equal number of black and white pixels, although each has a different shape than the others. So if Entropy is used as complexity measure criterion to select the appropriate network all of those blocks will be compressed by same network. Each of these blocks has different pattern and in order to obtain better compression ratio, it is better to assign these blocks to different networks. Therefore, we conclude that Entropy and Activity can not discriminate these different patterns. This causes a false selection of appropriate network in the algorithm and a poor estimation in true complexity approximation.

To overcome this problem we have used another complexity measure named *pattern-based* method. In this method image blocks are classified based on their patterns. This is done by dividing a block into four equal sub-blocks. The division method is based on quad-tree representation of an image, so the cross-cut of blocks is not considered. Then each image block is assigned to one of the 16 patterns of Fig. 3. The sub-blocks are black or white and it is necessary to use a threshold to assign black or white level to a grayscale sub-block. For each of these patterns one network is used to compress the associated image block. Networks have various compression rates based on their related patterns. That is, patterns number 1 and 2 in the Fig. 3 have maximum compression rates, 3 and 4 have the minimum compression rates. More details about the compression rates of these patterns are discussed in section IV.

IV. Experimental Results

In this section we have evaluated the compression ability of the basic network structure in Fig. 1 and proposed adaptive approach of Fig. 2 with different complexity measure criteria. Also we have compared the adaptive method with JPEG standard coding algorithm. In addition to Compression Ratio (CR) which is given in equations (4) and (5), the performances of these methods are compared according to

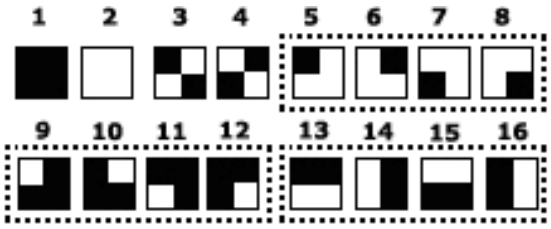


Fig. 3 Patterns used for classification of image blocks as a complexity measure criterion

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

Peak Signal to Noise Ratio (PSNR) criterion. PSNR is mostly used for its simplicity in calculation as a criterion to express the image quality generated by a lossy compression like the neural network-based method. Regardless to its simplicity, this method does not specifically related to the resulting compressed image quality as observed by a human. In this metric, the original image X is assumed as a clean signal which its de-compressed image, $\hat{X}$ is considered to depict the noisy signal. The original and de-compressed images are assumed to be of the same size. For an image of size $R \times C$ (i.e. $Row \times Col$) PSNR is determined according to equation (8) in decibel.

$$PSNR = 10 \log_{10}\left(\frac{255^2}{\frac{1}{R \times C}\sum_{i=1}^{R}\sum_{j=1}^{C}(X_{ij} - \hat{X}_{ij})^2}\right) \ (dB) \qquad (8)$$

We have used 8 bits/pixel grayscale images in our experiments, so 255 indicates the maximum gray-level in the above equation. The size of image blocks is one of the parameters which seem to need to be optimized. We have used 4×4, 8×8 and 16×16 as the size of image blocks which respectively result 16, 64 and 256 neurons in input and output layers. The evaluation of block size with above values is done in the basic structure of Fig. 1. In all of these networks fix compression ratio 4:1 (i.e. 16:4, 64:16 and 256:64, respectively) are used that correspond to 4, 16 and 64 neurons in the hidden layer (i.e., 2×2, 4×4 and 8×8 blocks). Our simulation results showed that the value of this parameter is not so crucial, but there are some considerations about it. Larger block size results higher number of parameters and requires more training patterns. Also this leads to higher error variance and relatively better PSNR for compressing the images which are out of the training set. Obviously this is reversed for small block sizes. Speed is another parameter which seems to be affected by the block size. The number of blocks in a particular image decreases as the size of block increases, so it seems that the speed of algorithm improves with larger block size. But the following considerations reject this superficial reasoning.

Suppose an image with N pixels is divided into K_s blocks, each block in size $\sqrt{N_s} \times \sqrt{N_s}$. In the basic compression structure and with CR 4:1, there are K_s times that a vector in size $N_s \times 1$ multiplies with the weight matrix of compressor in size $N_s \times \frac{N_s}{2}$. Also multiplying $\sqrt{\frac{N_s}{2}} \times \sqrt{\frac{N_s}{2}}$ compressed blocks in the de-compressor $\frac{N_s}{2} \times N_s$ weight matrix should be considered. So the number of multiplications and summations are as equation (9) and (10), respectively.

$$R_s^m = K_s\left[\left(N_s . \frac{N_s}{2}\right) + \left(\frac{N_s}{2} . N_s\right)\right] = N . N_s \qquad (9)$$

$$R_s^s = K_s\left[\left((N_s - 1) . \frac{N_s}{2}\right) + \left((\frac{N_s}{2} - 1) . N_s\right)\right] = N . (N_s - \frac{3}{2}) \qquad (10)$$

Where $K_s = \frac{N}{N_s}$. Now if a larger block size, $\sqrt{N_l} \times \sqrt{N_l}$ is used where $N_l = 2 . N_s$, then the equations (9) and (10) are changed as bellow.

$$R_l^m = N . (2 . N_s) = 2 . R_s^m \qquad (11)$$

$$R_l^s = N . (2 . N_s - \frac{3}{2}) = 2 . N . (N_s - \frac{3}{4}) \approx 2 . R_s^s \qquad (12)$$

It means that no improvement in speed is obtained by using larger block size. In general, the size of image block is not very critical parameter and among the three experienced block sizes we have used 8×8 in all of the experiments in this work.

A. Evaluations on basic network structure

To evaluate the basic network structure we used fix CR 4:1. Input image blocks are 8×8, a pixel has 8 bits, and 8 bits/neuron is assumed to code the outputs of the hidden layer. For training the network, we have used the Lena image of size 256×256 and no overlapping between image blocks is used. Lena with three another images are used as test images. Fig. 5 illustrates the result of CODEC with this structure and Table I shows the PSNR obtained by comparing the original images with their compressed version. BPNN results better performance for the trained image compared to other images in the test set. Clearly, this method has not provided acceptable quality for test images that are not in the training set. This fact is due to data dependency problem in extracted transform resulted by trained weights. This basic structure estimates PCA transform just for a trained image that is optimum only for this image.

There are some considerations in the implementation of the basic structure that can affect the results. The initialization of network parameters, number of quantization bits in hidden layer and the amount of training data are some important concepts which need more exploration. The initial weights for neural network structure are the starting points of the search in finding optimum transform. In these experiments we have used random initialization but there are other works like [3] which examine other values. They have used a feed-forward 3-layered network as CODEC and experimented DCT values as initial weights. Their evaluation has showed that this initialization does not bring about any improvement in de-compressed image quality. The number of quantization bits in the hidden layer affects the compression ratio and the reconstructed image quality. As mentioned, we have used 8 bits per hidden neuron output. In [3] the quantization level was varied from 1 to 10 bits and in [4] this value was changed between 1 and 8. Almost same result was achieved in both and the performance does not significantly increase in using more than 6 bits, while this improves the compression ratio. Although there is no deterministic rule for the amount of training data in parameter tuning for neural networks and the experience is the better teacher, but as a rule of thumb for each unknown parameter, 4 to 10 training samples are required. We considered this in our experiments. Generally, increasing the size of the training set results in increasing in

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf



Fig. 4 Test set images (Lena is also in training set)



Fig. 5 Reconstructed test set images using network of Fig. 1 for CODEC

learning error, but decreasing test error. As the training samples are increased, these two error values (i.e. learning error and test error) are converged to the same value.

B. Results of adaptive methods

In order to improve the performance of the basic network structure we have used the adaptive approach described in section II part C. This method needs more training data due to higher number of network parameters. Selection of the training set is done in such a way that a wide range of the complexities in images have been covered. Also the number of training patterns for all networks should be sufficient to achieve the convergence with minimum error. We have selected 105 images which contains a wide range of complexities for all gray-levels. Fig. 6 shows some samples of our training images.

TABLE I
COMPRESSION RESULTS USING BASIC EBPNN STRUCTURE

Test image	PSNR (dB)
Lena	34.92
Camera man	26.67
Crowd	23.41
Pepper	22.20

Like the basic structure, in training the adaptive networks, 8×8 image blocks are used, i.e. 64 neurons in the input layers. According to each complexity measure method, each block is given to its appropriate network. The blocks with low level of complexity are given to networks with higher compression ratio and those with high level of complexity are given to networks with lower compression ratio. In these structures the number of neurons in hidden layer determines the compression ratio and strictly is related to the complexity value of blocks. We have chosen this value for each network in each complexity criterion as discuss in the following.

Having the trained networks, given an input test image, the compression is done as the training routine. The images are divided into blocks and the complexity value for each block is computed. After that, the network related to that complexity value is selected and the compression is done by that network. We have called this method as *complexity-based* in our results in the following.

Another choice in selecting a network from among trained networks is to choose a network which is optimum for compressing an image block. In this approach the complexity measure is not considered and a network is selected such that it minimizes an error measure criterion. We have used this network selection method by considering "maximization of signal to noise ratio" criterion for each block. Here, an image block is given to all networks and the network which results maximum SNR for the block is selected. We named this approach *best-SNR* as shown in our results in the next sections. Of course, this is a time consuming process; however it results the minimum error in reconstructed image. It is clear that if the complexity measure criterion has been chosen precisely, it would have been expected that the results of these two approaches were the same or close together.

Also, in order to reduce the chess-board effect in reconstructed image and improve its visibility quality, overlapping the adjacent image blocks is allowed. We have realized this idea by considering a %50 overlap between rows and columns of neighboring blocks. In the overlapped area, for each pixel, the average value of overlapped pixels is calculated. As the results of the following sections show, overlapping improved image visibility quality and increased its PSNR.

For all three adaptive methods, the Lena image is presented in the training set and the other three images of Fig. 4 are not. The stopping conditions for Back-Propagation learning in all

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf



Fig. 6 Some sample images of the training set used in the adaptive approach.

three adaptive structures (i.e. Entropy, Activity and Pattern-based) have been both the maximum number of iterations 10000 and the error threshold $\varepsilon = 1.0e-5$ for equation (3).

i. Experimental results using Entropy measure

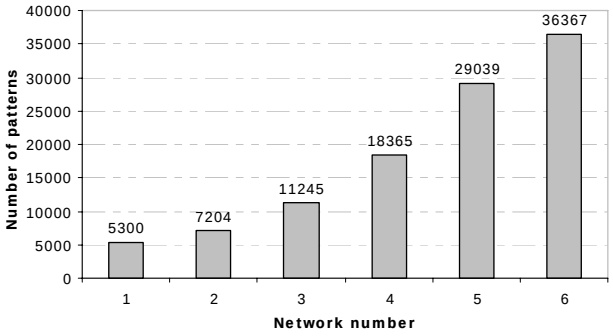
In Entropy complexity measure criterion, six networks are used which results 3 bits overhead for coding the number of networks (q in Eq. (5)). The numbers of hidden neurons for these networks are 4, 9, 16, 25, 36 and 49 that result 16, 7.1, 4, 2.5, 1.7 and 1.3 compression ratios, respectively. Of course, there are other possibilities in selecting the number of hidden layer neurons but in all of them higher complexity needs higher number of neurons and vice versa. An important notice in choosing these numbers is that large number of hidden neurons not necessarily leads to small error for test set, although it results small error in training data. Regards to the available range for gray-level values, 0 to 255, the maximum Entropy value will be $log_2\ 256 = 8$. This maximum is achievable when all of the gray-levels have the same probably. In the other word, an image block receives its maximum value of Entropy when it contains the same number of all of the gray values in that block. For a 256 level image the maximum Entropy value of 8 is achievable only when that image block has at least 256 pixels but whereas we have used 8×8 image blocks, this maximum value will be $log_2\ 64 = 6$. We have used Eq. (13) to assign the blocks to the networks. This results values of network number in $1 \le NN \le 6$.

$$\begin{cases} NN = \lfloor Entropy \rfloor + 1 & ; Entropy \neq 6 \\ NN = 6 & ; Entropy = 6 \end{cases} \quad (13)$$

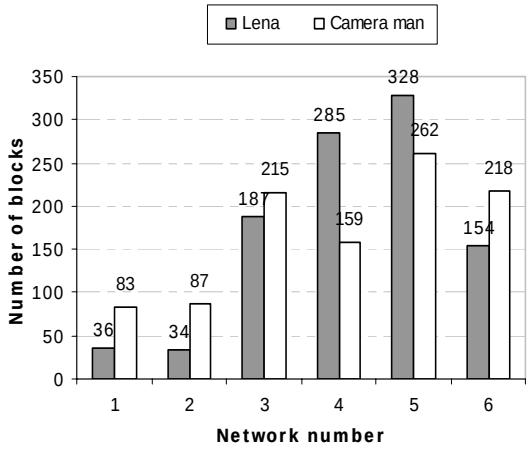
Also the selected number of hidden neurons is based on the fact that image blocks with higher Entropy should be coded with lower compression ratio network. Fig. 7 (a) shows the values of *NN* for all training images and Fig. 7 (b) shows this for Lena and Camera man images in the test set. The distribution of the number of patterns for each network in the training data, shows the sufficiently of data for training the networks.

In Entropy-based adaptive method the error threshold condition was satisfied before reaching the maximum number of iterations for convergence condition in BP learning. The results of this adaptive structure are shown in table II. The evaluation is done on 4 test images shown in Fig. 4 in Complexity-based and Best-SNR approaches with and without overlap for each image. The Compression Ratio (CR_a) is calculated using Eq. (5).

The ability of this method in good reconstruction of out-of-train images (i.e. images that are not present in training set) is considerable. The PSNR for all images of test set are close and due to the higher complexity of the Crowd image its CR is less than others. The reconstructed Lena images for experiments (a) to (d) of table II are illustrated in Fig. 8. As Fig. 8 (a) shows, the chess-board effect is evidence. Utilizing the block overlapping reduces this defect as shown in Fig. 8 (b) and increases the PSNR about 1dB without affecting the CR. The Best-SNR network selection for test blocks results



(a) [105 (256×256) images=107520 patterns]



(b)

Fig. 7 (a) The histogram of the Entropy values for all images in the training set and (b) for Lena and Camera man test images

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

TABLE II
COMPRESSION RESULTS BY ADAPTIVE BPNN STRUCTURE USING *ENTROPY* COMPLEXITY MEASURE CRITERION

Network Selection	Test image	PSNR (dB)	CR_a
(a) *Complexity-based None-Overlapped*	Lena	35.50	2.76
	Camera man	35.11	2.83
	Crowd	37.92	1.71
	Pepper	35.62	2.20
(b) *Complexity-based Overlapped*	Lena	36.70	2.76
	Camera man	36.57	2.82
	Crowd	38.68	1.71
	Pepper	36.25	2.20
(c) *Best-SNR None-Overlapped*	Lena	43.10	1.79
	Camera man	39.24	2.04
	Crowd	38.11	1.70
	Pepper	38.06	1.78
(d) *Best-SNR Overlapped*	Lena	44.55	1.79
	Camera man	41.04	2.06
	Crowd	39.73	1.69
	Pepper	39.71	1.74

TABLE III
THE AVERAGE COMPLEXITY (ENTROPY) FOR THE TEST SET IMAGES (256*256 IMAGES: 1024 BLOCKS IN SIZE 8*8)

Image	Crowd	Pepper	Lena	Camera man
Average Complexity	5.32	4.52	3.71	3.47



(a) (b) (c) (d)

Fig. 8 Reconstructed Lena image using Entropy-based adaptive compression approach with 4 different compression methods (a) to (d) from Table II

considerable improvement in PSNR value and visibility quality as parts (c) and (d) show in Table II and Fig. 8 (c) and Fig. 8 (d). From the other point of view, the drawbacks of this approach are increasing CODEC time and decreasing CR. The CR decline shows that the networks with lower CR are also selected to code the blocks with lower complexity. A remarkable note of results (a) to (d) in Table II is the distinction in PSNR improvement between Lena and Crowd images. This is because of difference in the complexity level of these images. In our test set the average complexity (Entropy) values are shown in Table III.

Lena and Camera man images are less complex than Pepper and Crowd. Crowd is the most complex one but using the Best-SNR has not provided significant improvement in the PSNR of its reconstructed image. This is because the improvement resulted by the Best-SNR method is due to the property of this method in using networks with lower CR for less complex blocks in an image. The fact is that, in Crowd image, 85% of the blocks have the complexity relate to $NN = 6$ and only 5% of them relate to $NN \leq 4$. This means that we could use networks with lower CR only for 5% of blocks. So, less degradation in the CR of this image compare to other images can be also justified

ii. Experimental results using Activity measure

In this method 4 networks with the same input block size as Entropy method are used. The networks have 9, 16, 25 and 36 hidden neurons that result 7.1, 4, 2.5 and 1.7 CRs, respectively. Theoretically the maximum value of the Activity in Eq. (8) for a 256 level 8×8 image block is higher than 3.0e+6 but this value for our training set is about 1.9e+6 which there are very small number of blocks that have high Activity near to this value. We have selected 4 different ranges [0,465), [465, 4073) and [4073, 20154) and [20154, ∞), each for a network. This is done regards to all Activity values in training set. Fig. 9 shows the number of patterns in training set for these 4 networks.

Table IV shows the results of this adaptive method. The resulted PSNR in this method is better than Entropy for less complex images in (a) and (b) tests. These results show that this measurement method estimates the complexity better than Entropy only for less-complex images.

The smaller number of networks in this method causes that the Best-SNR does not provide PSNR as good as Entropy-based method. On the other hand, these numbers of networks caused only 2 bits overhead compared to 3 bits in Entropy-based method. Also regards to the selection of number of neurons in the hidden layer, this method provides better CR. The results of this method in case (d) indicates that Best-SNR, for complex images, Crowd and Pepper, do not show high improvement. This is because the complexity of these images and same reasoning as previous section is correct about them.

iii. Experimental results using Pattern-based measure

The simulation of this method is done using 16 networks, one network for each pattern in Fig. 3. The CRs related to these patterns are 16:1 for patterns number 1 and 2, 4:1 for patterns numbers 6~8 and 9~12, 7.1:1 for patterns number 13~16 and 2.5:1 for patterns 3 and 4. Selection of the number of hidden layer neurons is based on the visual and intuitive

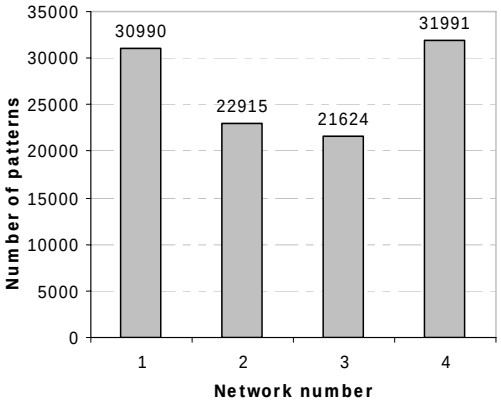


Fig. 9 The histogram of number of training blocks for 4 Activity values range for 105 images in training set. (i.e. Number of patterns= 105× (256×256: images size) =107520)

TABLE IV
COMPRESSION RESULTS BY ADAPTIVE BPNN STRUCTURE USING *ACTIVITY* COMPLEXITY CRITERION

Network Selection	Test image	PSNR (dB)	CR_a
(a) *Complexity-based None-Overlapped*	Lena	42.04	11.94
	Camera man	37.79	9.25
	Crowd	36.15	4.78
	Pepper	34.28	6.93
(b) *Complexity-based Overlapped*	Lena	43.12	11.79
	Camera man	38.24	9.16
	Crowd	36.41	4.75
	Pepper	34.93	6.67
(c) *Best-SNR None-Overlapped*	Lena	43.99	7.96
	Camera man	38.56	7.47
	Crowd	36.58	4.46
	Pepper	35.43	4.82
(d) *Best-SNR Overlapped*	Lena	45.09	7.93
	Camera man	39.04	7.38
	Crowd	36.82	4.38
	Pepper	35.74	4.81

TABLE V
COMPRESSION RESULTS BY ADAPTIVE BPNN STRUCTURE USING *PATTERN-BASED* COMPLEXITY CRITERION

Network Selection	Test image	PSNR (dB)	CR_a
(a) *Complexity-based None-Overlapped*	Lena	39.11	6.67
	Camera man	34.44	6.69
	Crowd	30.66	5.11
	Pepper	32.01	6.11
(b) *Complexity-based Overlapped*	Lena	40.20	6.67
	Camera man	34.80	6.73
	Crowd	32.00	5.10
	Pepper	32.86	6.19
(c) *Best-SNR None-Overlapped*	Lena	44.00	3.24
	Camera man	38.60	3.70
	Crowd	36.91	2.90
	Pepper	36.08	3.00
(d) *Best-SNR Overlapped*	Lena	46.76	3.26
	Camera man	40.45	3.71
	Crowd	39.41	2.89
	Pepper	38.14	3.01

methods to select these numbers. We can assume the compressed patterns in the hidden neurons as a smaller version of the original patterns and use the fix 64:16 CR for all networks. The CODEC results of this method are shown in Table V.

These results are almost in the same direction of two former methods. The overlapping increases the PSNR about 1dB and Best-SNR network selection method has resulted higher quality reconstructed images even for complex images. This is because of the higher number of networks used in this method compared to previous methods. Also due to the network structures the CR of images is higher than Entropy-base structure and is lower than Activity-based. There are 4 overhead bits in this method to code the network number. In addition, the reduction of CR value with Best-SNR is reasonable because of the higher number of networks.

C. Comparison of Entropy, Activity and Pattern-based methods

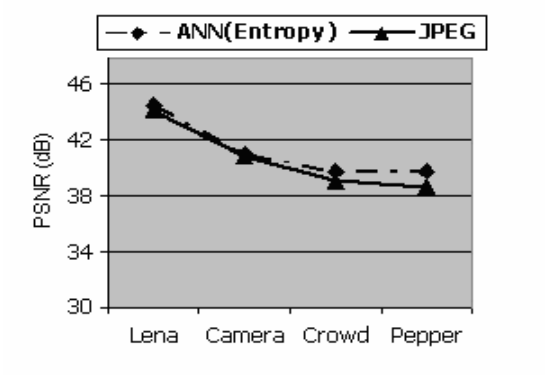
In the following we have compared three adaptive methods together and then compared these methods in their best case, Best-SNR with overlapping and with JPEG standard. The compression of three proposed adaptive methods using their results is not so reasonable because each one uses different numbers of networks and different structures in each network. In any compression approaches rate distortion or the tradeoff between the CR and data distortion is an important subject. In ANN-based adaptive approach it is possible to estimate the rate distortion function (RDF) theoretically and it can be done by making all possible networks with any CRs. Having RDF for each compression method enables us to compare various methods more precisely. Estimation of RDF is not performed for proposed methods in this research and the results are used to compare methods. As the results of Table III, IV and V show, the Entropy presents good estimation of complexity for complex images. On the other hand Activity performs better estimation for less-complex and simpler images. This method has resulted higher CRs due to its network structure. The Pattern-based complexity measure method has result almost the same PSNR with higher CR than Entropy and lower CR than Activity.

For better comparison of these methods and considering both CR and PSNR together, we have compared them with JPEG standard. The results are shown in Fig. 10. It illustrates the ability of adaptive compression methods compared to the standard JPEG algorithm. This comparison is done in the same bit rate for each method. These results show the achievements and even superiority of ANN-based compression to this compression standard.

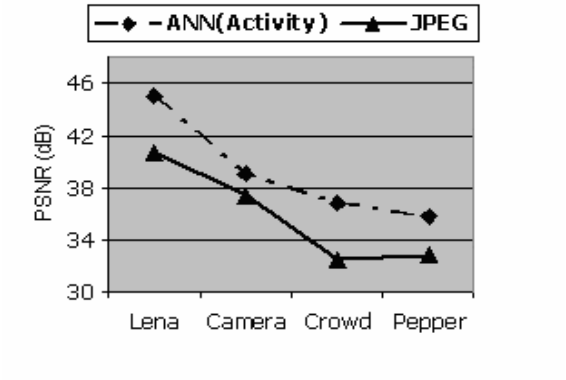
In addition to the compression results, the three proposed adaptive methods are different from fundamental principle point of view. Entropy is a measure of uncertainty of a random variable which quantifies the amount of information of a source, like an image. The Activity is an intuitional method for estimating the complexity of an image block using the difference between each pixel and its neighbors. These two methods do not discriminate the place or direction of the complexity and give an average value of complexity. For

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

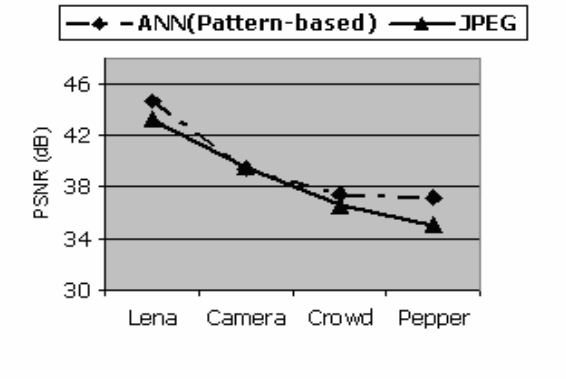
example the complexity value that is measured by Entropy or Activity for patterns number 15 and 16 in Fig. 3 are same, whiles these two patterns are completely different in ANN learning point of view. On the other hand, Pattern-based method classifies the image block into some predefined patterns. Actually, this method is not a complexity measure criterion and has not the mentioned problem of other methods, but this method indicates another problem. It does not consider grayscale values exactly and finally maps each 4×4 sub-blocks into one block or white pattern. One solution to this problem is the combination of Pattern-based method with other methods.



(a)



(b)



(c)

Fig. 10 The comparison of the adaptive methods with the JPEG compression standard: (a) Entropy, (b) Activity and (c) Pattern-based

V. Summary and Conclusion

We have reviewed the use of Multi-Layer Preceptoron Neural Networks for image compression. Since acceptable result is not resulted by compression with one network, an adaptive approach is used. It uses different networks for different image blocks regarding to their complexity values. Three complexity measurement methods Entropy, Activity and Pattern-based are presented and evaluated. Our experimental results show that better visual quality is obtained by overlapping neighboring image blocks. Also selecting images with Best-SNR criterion rather than the complexity criterion provides higher image quality and better PSNR. Higher number of networks provides better performance in Best-SNR approach but this will result in lower CR. However, overlapping and network selection need more investigations and it can be accepted to obtain better reconstructed image quality. Comparing results with standard JPEG algorithm shows better performance for our method both with PSNR measure and visibility quality.

In this paper the numbers of networks that provide different compression ratios are not optimized. It is expected that using larger number of networks and selecting optimum compression ratio for networks, provides better results. For this purpose some type of neural networks such as cascade-correlation can be used in addition to the heuristic or try and error approaches. In these networks we can select the number of neurons in hidden layer in such a way that an optimum compression ratio could be achieved. Among three proposed methods the Entropy and Activity do not use the orientation of patterns and the Pattern-based does not use the gray-levels properly. Considering the gray-level values in the Pattern-based method, can provide better results. This can be realized by combining the Activity or Entropy with Pattern-based. In addition, from a compression method viewpoint, the rate-control ability or having rate-distortion function is an important factor. However, it seems that ANN-based methods are not flexible in controlling the compression ratio, but it is possible to have a set of trained networks with different compression ratios rather than one network in each case.

References

[1] R. C. Gonzales, R. E. Woods, *Digital Image Processing*, Second Edition, Prentice-Hall, 2002.
[2] Veisi H., Jamzad M., Image Compression Using Neural Networks, Image Processing and Machine Vision Conference (MVIP), Tehran, Iran, 2005.
[3] N.Sonehara, M.Kawato, S.Miyake, K.Nakane, Image compression using a neural network model, International Joint Conference on Neural Networks, Washington DC, 1989.
[4] G.L. Sicuranza, G. Ramponi, S. Marsi, Artificial neural network for image compression, Electronic letters 26, 477-479, 1990.
[5] S. Marsi, G. Ramponi, G. L. Sicuranza, Improved neural structure for image compression, Proceeding of International Conference on Acoustic Speech and Signal Processing, Torento, 1991.
[6] S. Carrato, G. Ramponi, Improved structures based on neural networks for image compression, IEEE Workshop on Neural Networks for Signal Processing, New Jersey, September 1991.
[7] S. Carrato, S. Marsi, Compression of subband-filterd images via neural networks, IEEE Workshop on Neural Networks for Signal Processing. August 1992.

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf

[8] G. Qiu, M. Varley, T. Terrel, Image compression by edge pattern learning using multilayer perceptron, Electronic letters, Vol 29, No 7, April 1993.
[9] R.Sentiono, G. Lu, Image compression using a feedforward neural network, International Conference on Neural Networks, 1994.
[10] J. Jiang, Image compression with neural networks -A survey, Image Communication, ELSEVIER, Vol. 14, No. 9, 1999.
[11] C. Cramer, Neural networks for image and video compression: A review, European Journal of Operational Research, Vol. 108, July 1998.
[12] B. Verma, M. Blumenstein, and S. Kulkarni, A Neural Network Based Technique for Data Compression, Proceedings of the IASTED International Conference on Modelling and Simulation, MSO97, Singapore, 1997.
[13] A.Namphol, S.Chin, M. Arozullah, Image compression with a hierarchical neural network, IEEE Trans. Aerospace Electronic Systems Vol. 32 No.1, January 1996.
[14] J. S. Lin, S.H. Liu, A competitive continuous Hopfield neural network for vector quantization in image compression, Engineering Applications of Artificial Intelligence, Vol. 12, 1999.
[15] G. Pavlidis, A. Tsompanopoulos, A. Atsalakis, N. Papamarkos, C. Chamzas, A Vector Quantization – Entropy Coder Image Compression System, IX Spanish Symposium on Pattern Recognition and Image Processing, 2001.
[16] C. Amerijckx, J. D. Legaty, M. Verleysenz, Image Compression Using Self-Organizing Maps, Systems Analysis Modeling Simulation Vol. 43, No. 11, November 2003.
[17] S. Costa, S. Fiori, Image compression using principal component neural networks, Image and vision computing, Vol. 19, 2001.
[18] M. Egmont-Petersen, D. de Ridder, and H. Handels, Image processing with neural networks - a review, Pattern Recognition, vol. 35, pp. 2279-2301, 2002.
[19] A. Rahman and Chowdhury Mofizur Rahman, "A New Approach for Compressing Color Images using Neural Network", Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation - CIMCA 2003 , Vienna, Austria, 2003.
[20] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory, John Wiley and Sons Inc., New York, N.Y., 1991.
[21] Sonja Grgic, Marta Mrak, Mislav Grgic, Comparison of JPEG Image Coders Proceedings of the 3rd International Symposium on Video Processing and Multimedia Communications, VIPromCom, pp. 79-85, Zadar, June 2001.



Hadi Veisi received his MS in computer engineering (artificial intelligent) from Sharif University of Technology, Tehran, Iran in 2005. He is currently involving his Ph.D. in the same department. His research interests are Digital Signal Processing, Image Processing, and Neural Networks.



Mansour Jamzad Received his MS in Computer Science from McGill University in Montreal, Canada and his PhD in Electrical Engineering from Waseda University, Tokyo, Japan in 1989. He was a post doctorate researcher in Department of Communication and Electronic Engineering, Waseda University, Tokyo, from 1989 to 1990. He was a visiting researcher in Tokyo metropolitan institute of gerontology, during 1991-1994. He has become an assistant professor in Dept. of Computer Eng., Sharif University of Technology, Tehran, Iran, since 1995. His main research interests are Image processing, Machine Vision and Robotics.

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/4800/pdf