Journal of Signal Processing, Vol. 5, No. 6, pp. 445–454, November 2001

PAPER

Image Compression with Neural Networks Using Dynamical Construction Algorithm

Md. Imamul Hassan Bhuiyan[1], Md. Kamrul Hasan[1], Nait Charif Hammadi[2] and Takashi Yahagi[3]

[1]Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh
[2]Dept. de Genie Elect., Ecole Superieme de Tech., Univ. Mohammed I, Morocco
[3]Department of Information and Computer Sciences, Chiba University, Japan
E-mail: khasan@eee.buet.edu

Journal of Signal Processing

信号処理

Journal of Signal Processing, Vol. 5, No. 6, pp. 445–454, November 2001

PAPER

Image Compression with Neural Networks Using Dynamical Construction Algorithm

Md. Imamul Hassan Bhuiyan[1], Md. Kamrul Hasan[1], Nait Charif Hammadi[2] and Takashi Yahagi[3]

[1]Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh
[2]Dept. de Genie Elect., Ecole Superieme de Tech., Univ. Mohammed I, Morocco
[3]Department of Information and Computer Sciences, Chiba University, Japan
E-mail: khasan@eee.buet.edu

Abstract **In this paper, a modular approach of still image compression using neural networks (NNs) is presented. The proposed NN-based approach is composed of eight modules of neural networks, where each module is trained so as to suit best a particular image cluster. A dynamical construction algorithm (DCA) is developed for building each module of neural network. The DCA eliminates the need for preassignment of hidden layer size. Moreover, the NN constructed using the DCA, termed as dynamically constructive neural network (DCNN), has optimal size that reduces the hardware requirement. Wavelet transform based sub-image block classification technique is adopted for partitioning images into eight image clusters. The image compression system using modules of DCNNs gives better peak signal to noise ratio (PSNR) for a given bit rate as compared to other recent methods. Computer simulation results are presented to demonstrate the effectiveness of the proposed technique.**

Keywords: image compression, dynamical construction algorithm, dynamically constructive neural network, wavelet transform

1. Introduction

Due to the limitation of storage and transmission capacity, the efficient digital representation of signals has become inevitable for a wide class of applications, such as, video conferencing, remote sensing via satellite, digital TV/HDTV broadcasting, computer tomography (CT), magnetic resonance imaging (MRI), computer communications and so on [1], [2]. The term efficient digital representation refers to data compression, i.e., the elimination of redundancy from the raw digital data. The more correlated is the data, the more data items can be removed, resulting in larger degree of compression.

Numerous methods have been proposed so far for the lossy-compression of digital images [3]-[9]. Most current approaches fall into one of three major categories: predictive coding, transform coding, or vector quantization. In predictive coding technique, the difference between the true and its predicted image is coded for transmission. Since the magnitude of the error signal is less as compared to the true image, doing this requires less bits per second. However, the quality of the reconstructed image is highly dependent on the accuracy of the predictor.

The efficiency of the transform-based coders depends on the type of transform being used, e.g., the discrete cosine transform (DCT), or the wavelet transform. In the latter case, the appropriate selection of the wavelet for an arbitrary image is very important to obtain high compression ratio. The outcome of a vector quantizing image compression system is highly dependent on the choice of a code book. Besides, because of high computational complexity this technique is impractical for many applications.

Neural network based image compression is drawing much attention in recent days because of its massively parallel structure, high degree of interconnection and the propensity for storing experimental knowledge and capability to self-organize [7], [9]-[15].

Cottrell and Munro [10] used a two layer neural network for image compression. This method suffers from edge degradation at high compression ratio and restricted performance due to the difference between average intensities of training and test images [12]. This technique was extended to multilayer networks by Wahhab and Fahmy [14].

A multilayer hierarchical neural network based approach is developed in [13]. In [11], [12], [15], training images are divided into small sub-image blocks. Similar sub-image blocks are grouped to form image clusters which are used as training sets for different neural networks (NNs). This results in reduced edge degradation and improved generalization. However, all of these methods opt for an arbitrary selection of the size of the neural network.

A larger size network increases not only the learning time but also decreases its generalization ability [16]. On the other hand, a smaller size network may not learn completely the problem to be useful. As such the need for optimum selection of the network size is vital for applications such as image compression.

In this paper, we alleviate this problem by using dynamically constructive neural networks for image compression. A dynamical construction algorithm (DCA) is developed for building such a network. The dynamically constructive neural network (DCNN) built using the DCA provides nearly optimal size network and thereby eliminates the requirement for the preassignment of the network size. A modular approach is adopted for image compression using modules of DCNNs where each module of DCNN is optimized for a particular image cluster. Wavelet transform based sub-image block classification is adopted for partitioning training images into different image clusters.

This paper is organized as follows. In Section 2, we present neural image compression scheme. In Section 3, we describe the dynamical construction algorithm (DCA) for building a DCNN. In Section 4, we present wavelet based sub-image block classification technique. In Section 5, we present simulation results to show the efficacy of the proposed method. Finally, the paper concludes with some remarks in Section 6.

2. New Neural Image Compressor

Let that $\boldsymbol{X}$ denote the input image which is an $N \times N$ matrix of pixels given by

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NN} \end{bmatrix}$$

The whole image can be stretched to an $N^2 \times 1$ vector of pixels to supply as the input to a neural network consisting of N^2 input nodes. However, in case of digital images usually N^2 is a very large quantity and hence the size of the neural network. This requires a huge computational complexity which may be impractical in some cases. Alike the DCT based image compression techniques [17], we divide the input image into M^2 number of blocks of size $k \times k$ as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^1 & \boldsymbol{x}^2 & \cdots & \boldsymbol{x}^M \\ \boldsymbol{x}^{M+1} & \boldsymbol{x}^{M+2} & \cdots & \boldsymbol{x}^{2M} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}^{M^2-M+1} & \boldsymbol{x}^{M^2-M+2} & \cdots & \boldsymbol{x}^{M^2} \end{bmatrix}$$

where $kM = N$, and $\boldsymbol{x}^n$ is the nth block of the original image $\boldsymbol{X}$ and is given by

$$\boldsymbol{x}^n = \begin{bmatrix} x^n_{11} & x^n_{12} & \cdots & x^n_{1k} \\ x^n_{21} & x^n_{22} & \cdots & x^n_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ x^n_{k1} & x^n_{k2} & \cdots & x^n_{kk} \end{bmatrix}$$

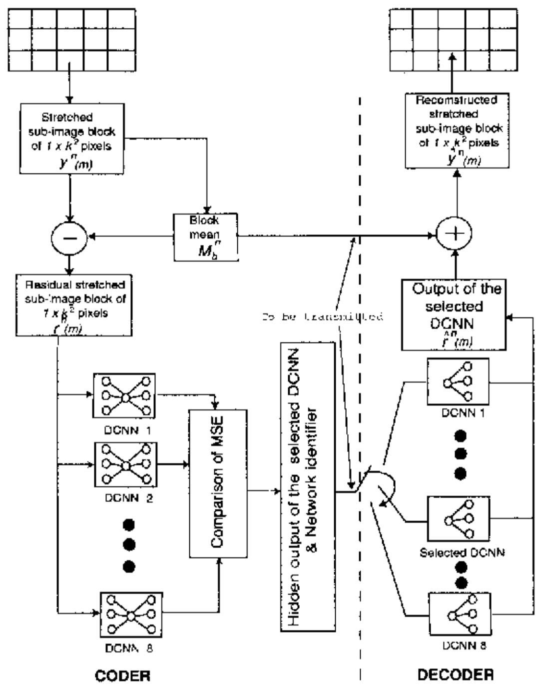


Fig. 1 Image compression using DCNNs

where x_{ij}^n denotes the value of the ijth pixel of the nth block. Each block is then stretched to a 1-D sequence given by

$$\begin{aligned}\boldsymbol{y}^n &= \left[y^n(1)y^n(2)\cdots y^n(k^2)\right]^T \\ &= [x_{11}^n\cdots x_{1k}^n x_{21}^n\cdots x_{2k}^n\cdots x_{k1}^n\cdots x_{kk}^n]^T \quad (1)\end{aligned}$$

During training, stretched sub-image blocks are classified into 8 image clusters and are used as training sets for 8 different DCNNs. The classification technique is described in Section 4. The DCNNs are trained in parallel on corresponding training sets. For example, a set of 8 DCNNs are trained on 8 different training sets. DCNN 1 is trained on training set 1, DCNN 2 is trained on training set 2 and so on. It is to be noted that the networks are trained on residual image blocks, obtained by subtracting block mean value from respective block pixels.

The proposed image compression scheme using DCNNs is shown in Fig. 2. The whole image is compressed block by block. As such, the input image is first divided into sub-image blocks of $k\times k$ pixels and then stretched to 1-D sequences of $1\times k^2$ pixels for feeding to DCNNs. Residual sequence of pixels $r^n(m)$ of the nth sub-image block are obtained by subtracting block mean from all the block pixels which are then fed into all the trained dynamical networks simultaneously.

The network with the least mean square error is selected as the desired block compressor. The uniformly quantized output of the hidden nodes of this network along with the mean value of the block (M_b^n) and the network identifier codeword constitute the compressed data for the nth sub-image block. An eight bit uniform quantizer is used to quantize the output of the hidden layer nodes as well as the block mean.

In the decompressor, the residual block pixel values ($\widehat{r}^n(m)$) are reproduced by linearly combining the quantized output of the hidden nodes of the selected network and the weights of this network from hidden layer to output layer. The reconstructed sequence of pixel values $\widehat{y}^n(m)$ are obtained by using the relation

$$\widehat{y}^n(m) = \widehat{r}^n(m) + M_b^n \quad (2)$$

where $m = 1, 2, \cdots, k^2$, k^2 is the total number of pixels in a block and $n = 1, 2, \cdots, M^2$.

3. Dynamical Construction Algorithm

So far mainly fixed architectures with a fixed number of hidden neurons have been considered for feedforward neural network. One of the few exceptions is the cascade correlation architecture proposed by Fahlman and its variants [18]- [21]. However, the cascade correlation network is not a simple multilayer network since each new neuron is itself a new layer and the number of layers is equal to the number of hidden neurons.

Thus, in the case of a large network the time necessary for an input to propagate to the output may be critical. One of the characteristics of the conventional single hidden layer network is its parallel processing architecture (i.e., the neurons in a hidden layer can be processed in parallel).

Therefore, the delay is minimal and it can be used in applications that require a quick response as in image compression-decompression problem. With this in mind here we propose a dynamical construction algorithm (DCA) that starts with a single hidden neuron and a new hidden neuron is added to the network whenever it fails to converge.

Before inserting the new hidden neuron into the network only the weights connecting the new hidden neuron to the other neurons are trained until there is a significant reduction of the output error.

The output o_j^p of the jth neuron in a two layer neural network is given by

$$o_j^p = f\left(\sum_{i=1}^{K} w_{ij} u_i^p + b_j\right) \tag{3}$$

where w_{ij} is the synaptic weight corresponding to the connection from the ith neuron in the previous layer to the jth neuron, u_i^p is the output of the ith neuron, K is the number of neurons that feeds the jth neuron (which is equal to the number of neurons in the previous layer), b_j is the bias, and f is the activation function. Usually, f is chosen to be a nonlinear function, e.g., sigmoid function. However, linear networks may outperform the nonlinear ones in terms of both training speed and compression performance [4], [9].

The hidden neurons are added dynamically one by one in a dynamically constructive neural network. Each new hidden neuron receives a connection from each of the network's inputs. All the *input-to-hidden* and ***hidden-to-output*** weights are trained repeatedly, not only the *hidden-to-output* weights.

The dynamical construction algorithm consists of cyclic repetition of three phases, *Train-Normal-Net* (denoted by P_{TNN}), *Train-Candidates* (P_{TC}), and *Neurons-Addition* (P_{NA}), after *Initialization* phase.

The *Initialization* phase initializes the parameters such as the learning rate, mean square error goal, number of epochs, the values of the weights and the maximum size of the neural network. P_{TNN} starts with a single hidden neuron and all the weights are trained with the back-propagation algorithm [22] which minimizes the mean-squared error (objective function) given by

$$E = \frac{1}{2P}\sum_{p=1}^{P}\sum_{i=1}^{I}(d_i^p - o_i^p)^2 \tag{4}$$

where P is the number of patterns in the training set, I is the number of neurons in the input and output layer, d_i^p is the desired output and o_i^p is the output of the ith neuron for the pth pattern.

One training cycle corresponds to the presentation of all the patterns in the training set to the network just once. After a given number of cycles T_N, we test whether desired error goal is achieved or the network has reached the predefined maximum size to stop the dynamical construction algorithm (DCA). If not, P_{TC} starts for further minimization of the residual error. In P_{TC} an independent *candidate* neuron is created (Fig. 2). This neuron is temporarily connected to all the input neurons and all output neurons of a virtual output layer.

The virtual output layer is a temporal layer of the same size as the original output layer (it has the same number of neurons as the original output layer). The output v_i^p of the ith neuron of the virtual layer is given by

$$v_i^p = f(Res_i^p + w_{ci}^c o_c^p) \tag{5}$$

where w_{ci}^c is the weight corresponding to the connection between the candidate neuron and the ith neuron of the virtual output layer, o_c^p is the output of the candidate neuron, and Res_i^p is given by

$$Res_i^p = \sum_{h=1}^{H} w_{hi} o_h^p + b_i \tag{6}$$

where H is the number of hidden neurons in this stage of the learning process, w_{hi} is the weight on the connection from the hth hidden neuron to the ith output neuron, and o_h^p is the output of the hth hidden neuron.

The output o_c^p of the candidate neuron is given by

$$o_c^p = f\left(\sum_{i=1}^{I} w_{ic}^c s_i^p + b^c\right) \tag{7}$$

where w_{ic}^c is the weight on the connection from the ith neuron of the input layer to the candidate neuron, s_i^p is the ith element of the input vector and b^c is the bias of the candidate neuron.

In the P_{TC}, all the previously trained weights are temporarily kept "frozen". For each candidate neuron, the *input-to-candidate* and *candidate-to-virtual* weights are trained to minimize the output error given by

$$E_{cand} = \frac{1}{2P}\sum_{p=1}^{P}\sum_{i=0}^{I}(d_i^p - v_i^p)^2 \tag{8}$$

P_{TC} stops when there is no significant reduction of the output error after T_{cand} cycle. Then

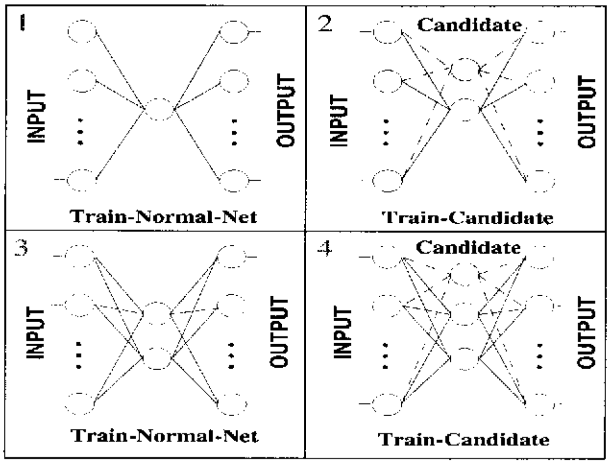


Fig. 2 Dynamical construction algorithm: A candidate neuron is created and the weights on dashed lines are trained while all others are kept frozen

P_{NA} starts, and the candidate neuron is definitely added to the network as a normal hidden neuron. And the whole network is then trained in P_{TNN}. This process is repeated until the maximum network size is reached or desired error goal is achieved.

The network obtained finally is called DCNN (dynamically constructive neural network). Fig. 2 presents an example where a network is being dynamically constructed. In the phase P_{TNN} (1 and 3) all the weights are updated. In the phase P_{TC} only the weights on the dashed connections are updated.

The advantage of the proposed algorithm is that it can find the size of the neural network without specifying it before training. Besides that, it results in a optimal network for a training set which may lead to better generalization [16]. This is augmented by the use of residual blocks for training of neural networks that removes the average intensity effect [12] and hence improves the generalization capability of the neural network.

4. Wavelet-Based Sub-image Block Classification Method

The wavelet transform decomposes a signal into a weighted sum of basis functions which are translated and dilated version of a prototype function ψ, known as wavelet prototype or mother wavelet. In the proposed method, the single level DWT (discrete wavelet transform) of all sub-image blocks are performed. In this work, we assume that the block size be 8×8, i.e., $k = 8$. Each image block is stretched to 1-D sequence of 1×64 pixels. For stretched image data $y^n(m)$, its DWT up to a level J is defined as

$$\begin{aligned} y^n(m) &= \sum_{j=1}^{J} \sum_{l \in Z} d_{2^j}(l) \psi^r_{2^j}(m - 2^j l) \\ &+ \sum_{l \in Z} a_{2^J}(l) \phi_{2^J}(m - 2^J l) \end{aligned} \tag{9}$$

where $\psi^r_{2^j}(m - 2^j l)$ are the analysis wavelets and $\phi_{2^J}(m - 2^J l)$ are the scaling sequences.

The detail coefficients, $d_{2^j}(l)$, express high frequency characteristics of the signal analyzed while the coarse or approximation coefficients, $a_{2^j}(l)$, express the low frequency characteristics of the signal analyzed. The detail and approximation coefficients are computed using Mallat's *herringbone algorithm* [23].

The coefficients are found by convolving $y^n(m)$ with appropriately designed qudrature mirror filters (QMF) and then downsampled by a factor of two. The QMF pair consists of a highpass filter with impulse function $c(m)$ and a lowpass filter with impulse function $g(m)$. The detail and approximation coefficients are then determined by the following equations:

$$d_{2^j}(l) = \sum_m y^n(m) c^*_{2^j}(m - 2^j l) \tag{10}$$

$$a_{2^j}(l) = \sum_m y^n(m) g^*_{2^J}(m - 2^J l) \tag{11}$$

Sub-image blocks are classified by their coarse energies. Coarse energy of a sub-image block represents its coarse level of activity and is defined as

$$\text{Coarse Energy}(CE) = \sum_{l=1}^{32} a_{2^j}(l)^2, \; j = 1 \tag{12}$$

Orthogonal wavelet, e.g., Daubechies wavelet [24] of order 3 is used as the mother wavelet. Coarse energies of all sub-image blocks are thus computed. The maximum of the computed energies is used to normalize the coarse energy of

Table 1 Threshold ranges of coarse energy

Class No.	Threshold Range
1	$0 < CE \leq 1.66$
2	$1.66 < CE \leq 3.33$
3	$3.33 < CE \leq 6.66$
4	$6.66 < CE \leq 8.33$
5	$8.33 < CE \leq 16.66$
6	$16.66 < CE \leq 33.33$
7	$33.33 < CE \leq 50$
8	$50 < CE \leq 100$

each block. The sub-image blocks are then arranged in ascending order of these energies and classified into eight groups or clusters of unequal elements by thresholding.

Table 1 presents the threshold ranges of coarse energy for different classes.

These eight groups are used as training sets for eight different DCNNs. Energy corresponding to detail coefficients is not taken into consideration for a sub-image block classification because the coarse energy is found to be quite dominant over it. However, any other feature from detail coefficients may be extracted for further improvement of the sub-image block classification technique. Research is undergoing in that direction.

5. Simulation Results

Three images, (each of 256×256 pixels with gray level intensity of 0-255), namely, trees, fruit and flowers were used for training. MATLAB [25] was used for simulation. The terms bit rate (BR), peak signal to noise ratio (PSNR) and mean square error (MSE), used in this paper for evaluating the performance of the proposed image compression scheme, are defined as follows.

The bit rate (BR) is given by

$$\mathrm{BR} = \frac{(Bh_q + s)}{64} \text{ bit/pixel} \tag{13}$$

where B is the number of bits used to code the signal at the hidden nodes, s is the side information (network identifier and block mean) to be transmitted and h_q is defined as

$$h_q = \frac{\sum_{h=1}^{64} h\beta_h}{\sum_{h=1}^{64} \beta_h} \tag{14}$$

where β_h is the number of blocks coded by a dynamical network with h hidden neurons.

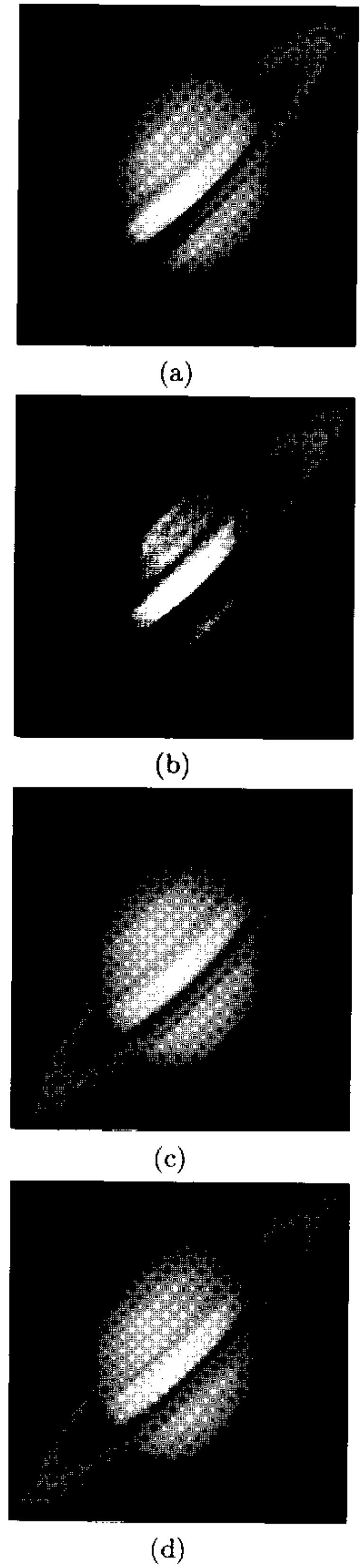


Fig. 3 Image compression results for 'Saturn' using DCNN: (a) Original 'Saturn' at 8 bit/pixel; (b) Reproduced 'Saturn' at 1.49 bit/pixel; (c) Reproduced'Saturn' at 0.70 bit/pixel; (d) Reproduced 'Saturn' at 0.53 bit/pixel

(a)

(b)

(c)

(d)

Fig. 4 Image compression results for 'Lena' using DCNN: (a) Original 'Lena' at 8 bit/pixel; (b) Reproduced image at 1.42 bit/pixel; (c) Reproduced image at 0.73 bit/pixel; (d) Reproduced image at 0.53 bit/pixel

The peak signal to noise ratio (PSNR) and mean square error (MSE) are calculated using Eqns. (15) and (16) defined as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\mathrm{dB} \qquad (15)$$

$$\mathrm{MSE} = \frac{1}{N^2}\sum_{n=1}^{M^2}\sum_{m=1}^{k^2=64}(y^n(m) - \widehat{y}^n(m))^2 \qquad (16)$$

where N^2 is the total number of pixels in the image and M^2 is the total number of blocks in the image.

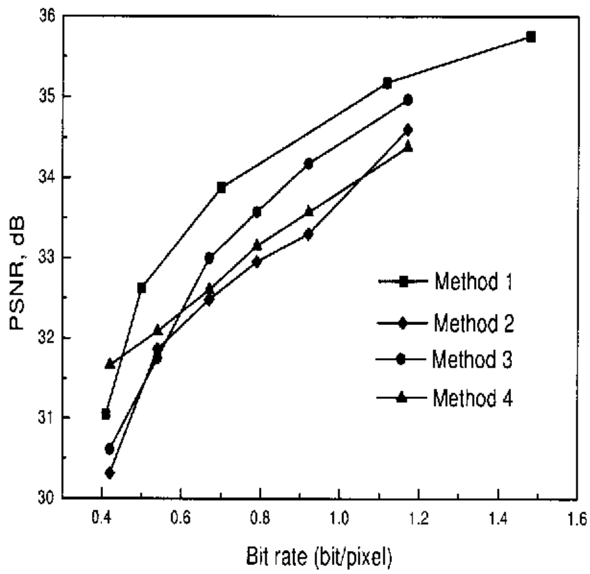
The performance of the proposed DCNN scheme is tested with the standard images 'Saturn' and 'Lena'. Notice that theses images are not used for training DCNNs. The reconstructed 'Saturn' and 'Lena' images using the proposed method are shown in Figs. 3 and 4, respectively.

The fidelity of the reproduced images using the proposed method is indeed quite satisfactory. In Fig. 5, a comparison of the different methods in terms of PSNR for different bit rates is presented for the test image 'Saturn'. In this figure, Method 1 is the proposed method (image compression using DCNN), Method 2 and Method 3 are described in [15] and Method 4 is described in [12].

Apart from a slight aberration at the beginning, the proposed method outperforms the other methods. The better performance of the proposed method may be attributed to the optimal size of a DCNN for a training set as it may lead to better generalization. As a result, the selected DCNN can reproduce the corresponding sub-image block with better generalization.

In contrast, in the methods described in [12] and [15], the number of nodes in the hidden layer are arbitrarily assigned. The neural networks are not optimal. Moreover, the classification technique adopted in this paper is different from that of [12] and [15]. The better performance signifies that similar image blocks are grouped in an image cluster to a greater degree as compared to the methods described in [12] and [15].

Another advantage of the proposed method is that it can provide optimal size network. This may help in reducing hardware requirement. For example, the number of nodes in the hidden layer of the 8 different neural networks that provide a



(a)

Fig. 5 Comparison of different methods in terms of PSNR against bit rate for test image 'Saturn'

PSNR of 34.66 dB (for test image 'Saturn') are 1, 5, 8, 5, 7, 7, 7, and 10.

On the contrary, the number of hidden nodes of all the 8 neural networks in other methods (Method 2, Method 3 and Method 4) is 7, producing the same PSNR. The reduction in the number of connections is approximately 10%. The number of nodes in the hidden layer of the 8 different neural networks that provide a PSNR of 27.55 dB for the test image 'Lena' are 1, 5, 6, 5, 6, 5, 6, and 10. On the other hand, the number of hidden nodes of all the 8 neural networks in other methods (Method 2, Method 3 and Method 4) is 8, producing the same PSNR. The reduction in the number of connections is approximately 29%.

6. Conclusions

We have presented a modular approach of image compression using neural networks built with dynamical construction algorithm (DCA). The DCA eliminates the requirement for the preassignment of the hidden layer size as required by most of the conventional methods. Each DCNN is trained on a particular image cluster. Wavelet transform based sub-image block classification technique is proposed for partitioning training images into different image clusters.

The image compression system using DCNNs demonstrate better PSNR for a given bit rate as compared to recently proposed image compression methods using neural networks. In addition, DCNN has optimal size that may reduce hardware requirement.

The proposed sub-image block classification technique considers coarse energy factor only. Inclusion of a feature parameter from detail coefficients and fuzzy decision making in thresholding of energies may provide better classification. Further work is being carried out in these directions.

References

[1] Milan Sonka, Vaclav Hlavac and Roger Boyle: Image Processing, Analysis and Machine Vision, pp. 621-622, PWS Publishing Company, 1999.

[2] R.C. Gonzalez and R. E. Woods: Digital Image Processing, pp. 307-309, Addison-Wesley Longman Pte Ltd., Singapore, 1999.

[3] A. K. Jain: Fundamentals of Digital Image Processing, pp. 189-190, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[4] S. Carrato: Neural networks for image compression, Neural networks:Adv. and Appl., 2 ed., pp. 177-198, Gelenbe Pub, North-Holland, Amsterdam, 1992.

[5] A. Gersho and R. M. Gray: Vector Quantization and Signal Compression, Kluwer, Norwell, MA, 1992.

[6] J. M. Shapiro: Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Processing, Vol. 41, No. 12, pp.3445-3462, 1993.

[7] R. D. Dony and S. Haykin: Neural network approaches to image compression, Proc. IEEE, Vol. 83, No. 2, pp.288–303, 1995.

[8] A. Said and W. A. Pearlman: A new, fast and efficient image codec based on set partitioning in hierarchial trees, IEEE Trans. CAS for Video Technology, Vol. 6, No. 3, pp.243-250, 1996.

[9] J. Jiang: Image compression with neural networks: A survey, Signal Processing - Image Communication, Vol. 14, No. 9, pp. 737-760, 1999.

[10] G. W. Cottrell and P. Munro: Principal components analysis of images via back propagation, Proc. SPIE-Int. Soc. Opt. Eng., Vol. 1001, pp.1070–1077, 1988.

[11] S. Carrato and S. Marsi: Parallel structure based on neural networks for image compression, Electronics Letters, Vol. 28, No. 12, pp. 1152-1153, June 1992.

[12] G. Qiu, M. R. Varley and T. J. Terrel: Image compression by edge pattern learning using multilayer perceptrons, Electronics Letters, Vol. 29, No. 7, pp. 601-602, April 1993.

[13] A. Namphol, S. H. Chin and M. Arozullah: Image compression with a hierarchical neural network, Trans. IEEE Aerosp. Electron. Syst., Vol. 32, No. 1, pp.326–337, 1996.

[14] O. A. Wahhab and M. M. Fahmy: Image compression using multilayer neural networks, Proc. IEE, Image Signal Process., Vol. 144, No. 5, pp.307–312, 1997.

[15] N. M. S. Rahim and T. Yahagi: Image compression by new sub-image block classification techniques using neural networks, IEICE Trans. Fundamentals, Vol. E83-A, No. 10, pp. 2040-2043, 2000.

[16] T. Chen: The past, present, and future of neural networks for signal processing, IEEE Signal Processing Magazine, pp.28–48, 1997.

[17] K. R. Rao and P. Yip: The Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press, New York, 1990.

[18] S. E. Fahlman and C. Lebier: The cascade-correlation learning architecture, Advances in Neural Information Processing Systems II, pp. 524-532. 1989.

[19] S. E. Fahlman and C. Lebier: Recurrent cascade-correlation architecture, Advances in Neural Information Processing Systems III, Morgan Kaufman Publishers, pp. 190–196. 1991.

[20] E. Littman and H. Ritter: Cascade network architectures, Proc. Int. Joint Conference on Neural Networks II, pp. 398–404 1992.

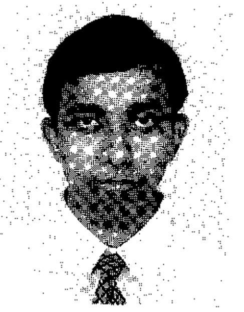
[21] E. Littman and H. Ritter: Cascade LLM networks, Artificial Neural Networks II, Elsevier Science Publishers, pp. 253-257, 1992.

[22] S. Haykin: Neural Networks: A Comprehensive Foundation, pp. 161-175, Prentice Hall, Englewood Cliffs, NJ, 1999.

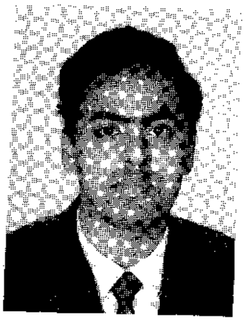
[23] S. Mallat: A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. PAMI, Vol. 11, No. 7, pp. 674-693, 1989.

[24] Yves Meyer: Wavelets Algorithms and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 1993.

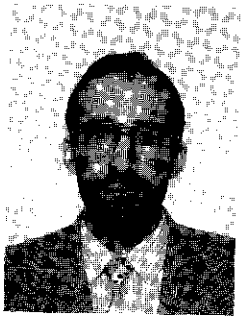
[25] MATLAB: The Mathworks Inc., 3 Apple Hill Drive, Natick, MA ; 01760-2098, USA.



Md. Imamul Hassan Bhuiyan received the B.S. and M.S. degrees in electrical and electronic engineering from Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh, in 1998 and 2001, respectively. Presently, he is serving as a Lecturer in the Department of Electrical and Electronic Engineering of BUET. His research interests are in digital signal processing and image compression. He is an associate member of IEEE.



Md. Kamrul Hasan received the B.S. and M.S. degrees in electrical and electronic engineering from Bangladesh University of Engineering & Technology (BUET), Dhaka, Bangladesh, in 1989 and 1991, respectively. In 1995, he received his M.Eng. degree in information and computer sciences from Chiba University, Japan. He received his Ph.D. degree in information and computer sciences in 1997 from the same university. In 1989, he joined as a Lecturer in the Department of Electrical and Electronic Engineering of BUET. From 1991 to 1993 he was a research student at Chiba University, Japan. From April 1997 to March, 2001, he was an Assistant Professor in the Department of Electrical and Electronic Engineering of BUET. Presently, he is an Associate Professor of the same department. His research interests are in digital signal processing, system identification, and image compression. He is a member of IEEE and IEICE (Japan).



Nait Charif Hammadi received the Ingenieur d'Etat diploma in Electronics and Control, from Ecole Hassania des Traveaux Publics, Casablanca, Morocco, in 1990. In 1991, he joined Ecole Superieure de Technologie, Mohammed I University, Morocco, as a Lecturer. In 1998, he received his Ph. D. in information and computer sciences from the Graduate School of Science and Technology, Chiba University, Japan. Then he rejoined Ecole Superieure de Technologie as Assistant Professor. From June to September, 1999, he was a Fulbright visiting scholar at Michigan State University, MI, USA. His research interests include neural networks, image processing, control and fault computing.

Takashi Yahagi received the B. S., M. S., and Ph. D. degrees in electronics from the Tokyo Institute of Technology, Tokyo, Japan, in 1966, 1968, 1971, respectively. In 1971, he joined Chiba University, Chiba, Japan, as a Lecturer in the Department of Electronics. From 1974 to 1984 he was an Associate Professor, and in 1984 he was promoted to Professor in the Department of Electrical Engineering. From 1989 to 1998, he was with the Department of Information and Computer Sciences. Since 1998 he has been with the Department of Information Science of the Graduate School of Science and Technology, Chiba University. His current research interests are in the theory and applications of digital signal processing and other related areas. He is the author of "Theory of Digital Signal Processing", Vols. 1-3 (1985, 1985, 1986), "Digital Signal Processing and Basic Theory" (1996), and "Digital Filters and Signal Processing" (2001), and the co-author of "Digital Signal Processing of Speech and Images" (1996), "VLSI and Digital Signal Processing" (1997), "Multimedia and Digital Signal Processing" (1997), "Neural Network and Fuzzy Signal Processing" (1998), "Communications and Digital Signal Processing" (1999), "Fast Algorithms and Parallel Signal Processing" (2000) (Corona Pub. Co., Ltd., Tokyo, Japan). He is Editor of the "Library of Digital Signal Processing" (Corona Pub. Co., Ltd., Tokyo, Japan). Since 1997, he has been President of the Research Institute of Signal Processing, Japan, and also Editor-in-Chief of the Journal of Signal Processing. Prof. Yahagi is a member of IEEE (USA), IEICE (Japan), The New York Academy of Sciences (USA), Japanese Society of Printing Science and Technology, etc.

(Received May 8, 2001; revised July 30, 2001)