Image Compression Technique Using a Hierarchical Neural Network

Rafid A. Khalil
College of engineering
University of Mosul

Mohammed C. Younis
Dep. Of Computer science
Al-Hadbaa University College

Received on: 13/11/2005 **Accepted on: 04/10/2006**

ABSTRACT

This paper present a Resilient Backpropagation (RBP) algorithm based on hierarchical neural network for image compression. The proposed technique includes steps to break down large images into smaller blocks for image compression/ decompression process. Furthermore, a Linear Backpropagation (LBP) algorithm is also used to train hierarchical neural network, and both training algorithms are compared. A number of experiments have been achieved, the results obtained, are the compression rate and Peak Signal to Noise Ratio $PSNR$ of the compressed/ decompressed images which are presented in this paper.

Keywords: Image compression, Resilient Backpropagation (RBP) algorithm, hierarchical neural network, Noise.

تقنية ضغط الصورة باستخدام شبكة عصبية هرمية

رافد خليل
كلية الهندسة، جامعة الموصل

محمد يونس
قسم علوم الحاسوب، كلية الحدباء الجامعة

تاريخ الاستلام: 2005/11/13 **تاريخ القبول: 2006/10/04**

الملخص

البحث يقدم خوارزمية الانتشار العكسي المرنة (RBP) مبنية على شبكة عصبية هرمية (أو تسلسلية) لضغط الصور. إن التقنية المقترحة تتضمن خطوات لتقطيع الصور الكبيرة إلى أجزاء صغيرة مناسبة لعمليات ضغط وفك الضغط للصور. فضلاً عن ذلك، استخدمت خوارزمية الانتشار العكسي الخطية (LBP) أيضا، لتدريب الشبكة العصبية الهرمية كلا الخوارزميتين RBP و LBP تمت مقارنة أدائهما من خلال إجراء عدد من التجارب. وإن النتائج المستخلصة مثل معدل الضغط وأعلى نسبة الإشارة إلى الضوضاء $PSNR$ قدمت في هذا البحث.

الكلمات المفتاحية: الضغط للصور، خوارزمية الانتشار العكسي المرنة، شبكة عصبية هرمية (أو تسلسلية)، الضوضاء.

1. Introduction

Image compression is a key technology in the development of various multimedia computer services and telecommunication applications, such as teleconferencing, digital broadcast codec and video technology, etc [5]. Traditional techniques that have already been identified for data compression include: predictive coding, transform coding and vector quantization [2].

Artificial Neural Networks ANNs have been applied in many problems; have demonstrated their superiority over traditional methods, when dealing with noisy data. One such application is for image compression. Neural networks seem to be well suited to this particular function, as they have the ability to preprocess input patterns to produce simpler patterns with fewer components [6]. This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. Not only can ANN based techniques provide sufficient compression rates of the data in question, but security is easily maintained. This occurs because the compressed data that is sent along a communication line is encoded and does not resembles its original form.

Several literatures were discussed the subject of applying ANNs to image compression in detail as in [2-5]. Many different training algorithms and architectures have been used. Some of those are: linear backpropagation [5], and sigmoidal back propagation [10], training two layer feed forward auto associative neural networks. Radial basis function RBF neural network [11], and self- organizing maps SOMs neural network, which after training acts as a code book [1].

The purpose of this paper is to implement a technique for compression of images using a Resilient Backpropagation RBP algorithm [8] based hierarchical feed forward auto associative neural network [5]. The method for training is faster than some of the more popular algorithms such as Backpropagation [9]. Results are presented for hierarchical neural network trained with both Resilient Backpropagation RBP algorithm, and Linear Back propagation LBP algorithm. The results convey information about the compression rate achieved, the Peak Signal to noise ratio PSNR, and a comparison of the images after decompression.

This paper is organized as follows: Section 2, states the RBP training algorithm. Section 3, discusses the hierarchical neural network architecture for image compression. Results are presented in section 4. A discussion of the results is supplied in section 5, and conclusions are drawn in section 6.

2. Resilient Back propagation RBP Algorithm

The conventional Backpropagation BP learning algorithm proposed by Rumelhart et. al. [9] uses the partial derivatives of the error function (i.e. the gradient) to minimize the global error of the neural network by performing a gradient descent.

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial E^{(t)}}{\partial w_{ij}} \qquad \text{...............} \ \mathbf{(1)}$$

The choice of the learning rate parameter η, which scales the derivative, has an important effect on the time needed until convergence is reached. As it can be easily observed, the size and the sign of the derivative are considered when updating the synaptic weights w_{ij}.

Resilient Backpropagation RBP algorithm is a local adaptive learning scheme, much faster and stable than other usual variations of the backpropagation algorithm. The basic principle is to eliminate the harmful and un-foreseeable influence of the size of the partial derivative on the weight step. So, only the sign of the derivative is used to indicate the direction of the weight update. The size of the update is given by a weight-specific update value Δ_{ij} [8]:

$$\Delta_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} \ , \text{ if } \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)} \ , \text{ if } \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0 \ , \text{ else} \end{cases} \qquad \textbf{................(2)}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \qquad \textbf{................ (3)}$$

The update values are determined from the following equation:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{-} * \Delta_{ij}^{(t-1)}, \text{ if } \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} * \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \eta^{+} * \Delta_{ij}^{(t-1)}, \text{ if } \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} * \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \Delta_{ij}^{(t-1)}, \text{ else} \end{cases} \qquad \textbf{.......... (4)}$$

where $0 < \eta^{-} < 1,\ \eta^{+} > 1.$ [8]

The RBP learning algorithm is based on learning by epoch; that means the weight update is performed only after the gradient information is completely available, after each training pattern has been presented and the gradient of the sum of pattern errors is known.

Some authors have noticed that when using standard back propagation [9], the weights in the hidden layer are updated with much smaller amounts than the weights in output layer, so they modify much slower. Another advantage of RBP is that all of the weights grow uniformly.

3. Hierarchical Neural Network for Image Compression [1]

The hierarchical neural network structure can be shown in Fig. (1), in which the three hidden layers are termed as the combiner layer, the compressor layer, and decompressor layer. The idea is to exploit correlation between pixels by inner hidden layer and to exploit correlation between blocks of pixels by outer hidden layers. From the input layer to the combiner layer and from the decombiner layer to the output layer, local connections are designed which have the same effect as M fully connected neural sub-networks. As seen in Fig. (1), all three hidden layers are fully connected. The basic idea is to divide an input image into M disjoint sub-scenes and each sub-scene is further partitioned into T pixel blocks of size $p \times p$. For a standard image of 256x256 pixels, it can be divided into 8 sub-scenes and each sub-scene has 128 pixel blocks of size 8x8. accordingly, the proposed neural network structure is designed to have the following parameters: total number of neurons at the input layer $= Mp^2$ = 8 x 64= 512. Total number of neurons at the combiner layer $= MN_h$ = 8 x 8= 64. Total number of neurons at the compressor layer = Q = 8.

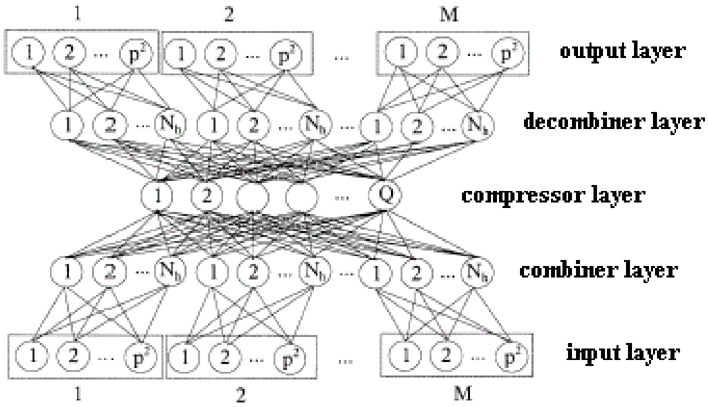


Fig. (1) Hierarchical neural network structure.

The total number of neurons for the decombiner layer and the output layer is the same as that of the combiner layer and the input layer, respectively. Training of such neural network can be explained as follows:

(i) Outer loop neural network (OLNN) training. By taking the input layer, the combiner layer and the output layer out of the network shown in Fig. (1), we can obtain M 64-8-64 outer loop neural networks where 64-8-64 represents the number of neurons for its input layer, hidden layer and output layer, respectively. As the image is divided into 8 sub-scenes and each sub-scenes has 128 pixels blocks each of which has the same size as that of the input layer, we have 8 training sets to train the 8 outer- loop neural networks independently. Each training set contains 128 training patterns (or pixel blocks). In this training process, the proposed RBP learning rule is directly applied in which the desired output is equal to the input.

(ii) Inner loop neural network (ILNN) training. By taking the three hidden layers in Fig. (1) into consideration, an inner loop neural network can be derived as shown in Fig. (2). As the related parameters are designed in such a way that $N_h = 8; \mathrm{Q} = 8; \mathrm{M} = 8$, the inner loop neural network is also a 64- 8- 64 network. Corresponding to the 8 sub-scenes each of which has 128 training patterns (or pixel blocks), we also have 8 groups of hidden layer outputs from the operation of step 1, in which each hidden layer output is an 8-dimensional vector and each group contains 128 such vectors. Therefore, for the inner loop network, the training set contains 128 training patterns each of them is a 64-dimensional vector when the outputs of all the eight hidden layers inside the OLNN are directly used to train the ILNN. Again, the RBP learning rule is used in the training process. Throughout the two steps of training for both ILNN and OLNN, the linear transfer function (or activating function) is used.

(iii) Reconstruction of the over all neural network. From the previous two steps of training, we have four sets of coupling weights, two out of step 1 and two out of step 2. Hence, the over all neural network coupling weights can be assigned in such a way that the two sets of weights from step 1 are given to the outer layers in Fig. (1) involving the input layer connected to the combiner layer, and the decombiner layer connected to the output layer. Similarly, the two sets of coupling weights obtained from step 2 can be given to the inner layer in Fig. (2) involving the combiner layer connected to the compressor layer and the compressor layer connected to the decombiner layer. After training is completed, the neural network is ready for image compression in which half of the network acts as an encoder and the other half as a decoder. The neuron weights maintained the same throughout the image compression process.

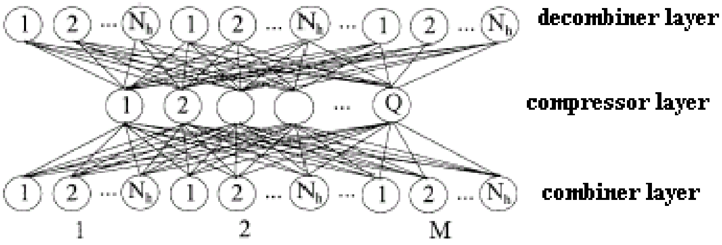


Fig. (2) Inner loop of hierarchical neural network.

4. Performance assessments

The performance of the hierarchical neural network trained by RBP and LBP learning rules can be assessed by considering the following measurements:

1. compression ratio [7], and can be defined for a narrow channel compression neural network as follows:

$$\textbf{Compression ratio} \equiv Mp^2 : \frac{qQ}{8} \qquad \textbf{............ (5)}$$

where q is the number of bits used to quantize each neuron output of compressor layer.

2. Peak Signal to Noise Ratio $PSNR$ [4], in lossy compression, the peak signal to noise ratio $PSNR$ is used as the measure of similarity or of dissimilarity, although it does not necessarily reflect visual quality. Assuming that the original and reconstructed images are represented by functions $f(x,y)$ and $g(x,y)$ of the pixel plane

position (x, y), respectively, the $PSNR$ is defined for 256-gray level image with T blocks of size p^2 pixels as follows:

$$PSNR = 10\log\frac{(256-1)^2}{e_{r.m.s}{}^2}, dB \qquad \text{………………(6)}$$

where the root- means square error is given by:

$$e_{r.m.s}{}^2 = \frac{1}{Tp^2}\sum_{x=1}^{T}\sum_{y=1}^{p^2}[g(x,y) - f(x-y)]^2 \qquad \text{………………(7)}$$

5. Experimental Results and Discussions

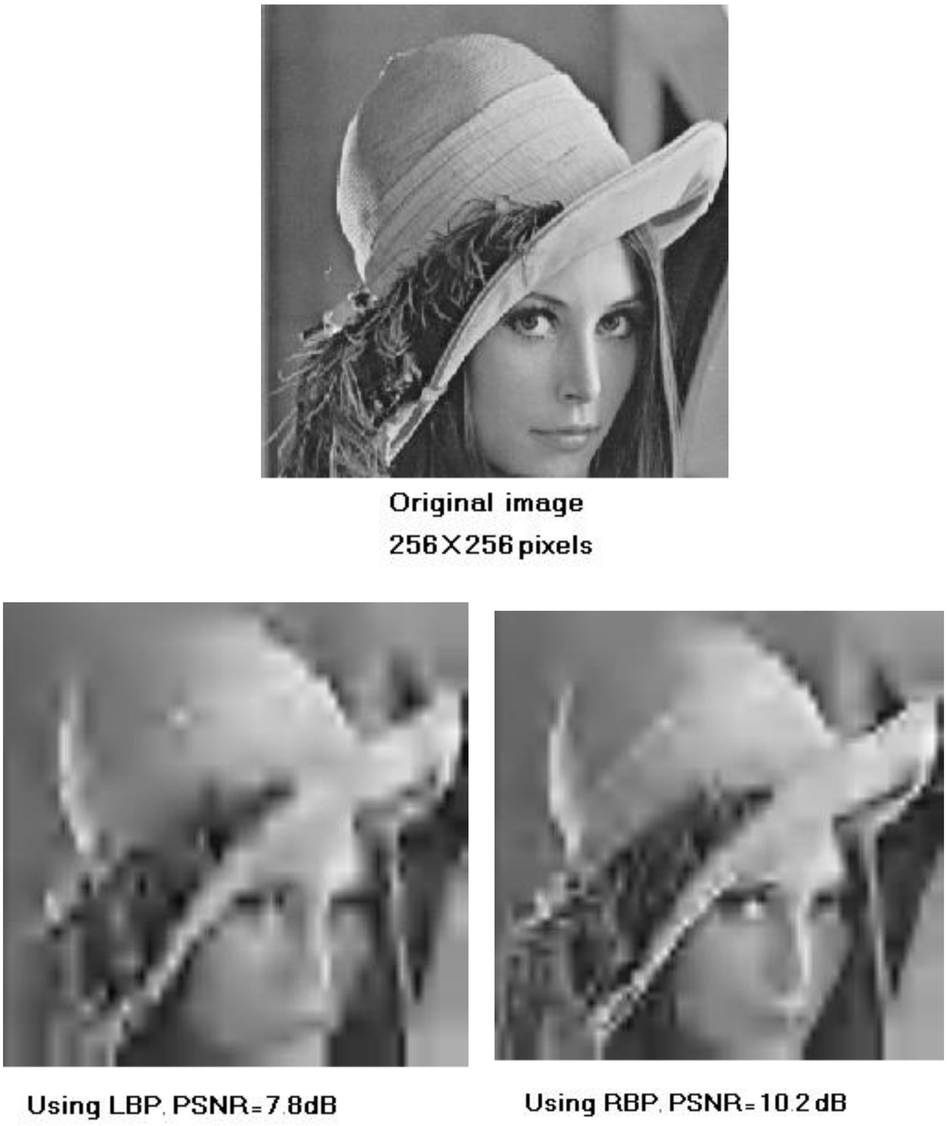
The RBP based hierarchical neural network from section 2, along with the LBP algorithm were tested, and the results were compared for the task of image compression. We take Lenna as one of most widely used image for testing image compression algorithms. Tables (1 and 2) summaries the experimental results, and Figures (3 to 6) show the compression performance for the RBP and LBP based hierarchical neural network for different network parameters.

Table 1. RPB based hierarchical neural network with Mp^2 = 512, and MN_h = 64, on Lenna image of (256x256) pixels, 256 gray level.

Compression layer neurons Q	Quantization level	Compression ratio	PSNR, dB
8	2^3	170 : 1	10.2
8	2^4	128 : 1	15.6
16	2^3	85 : 1	19.7
16	2^4	64 : 1	24.1

Table 2. LPB based hierarchical neural network with Mp^2 = 512, and MN_h = 64, on Lenna image of (256x256) pixels, 256 gray level.

Compression layer neurons Q	Quantization level	Compression ratio	PSNR, dB
8	2^3	170 : 1	7.8
8	2^4	128 : 1	12.7
16	2^3	85 : 1	17.2
16	2^4	64 : 1	22.3

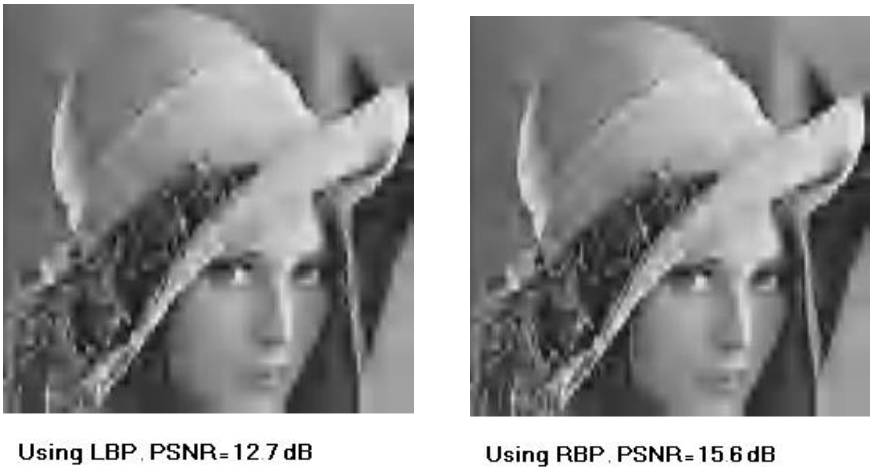


Original image
256X256 pixels

Using LBP, PSNR=7.8dB

Using RBP, PSNR=10.2dB

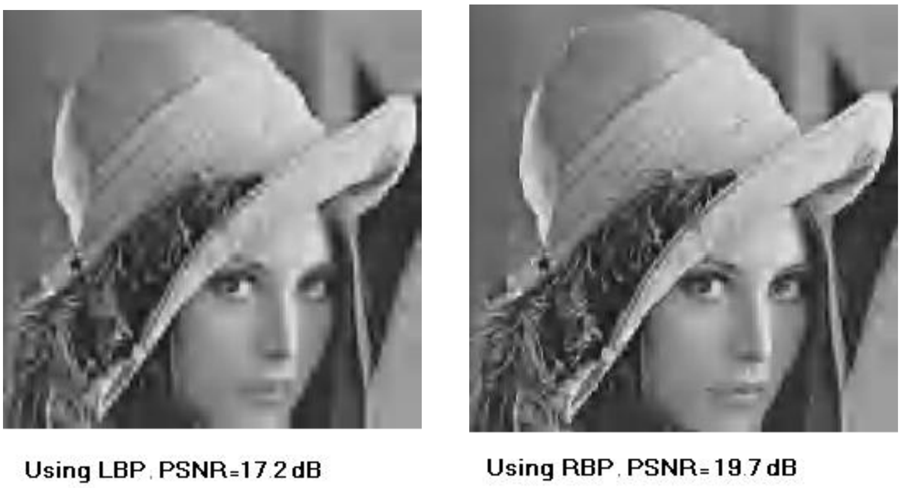
Fig. (3) Testing results for 170:1 compression ratio with Q= 8 neurons.



Using LBP, PSNR=12.7 dB

Using RBP, PSNR=15.6 dB

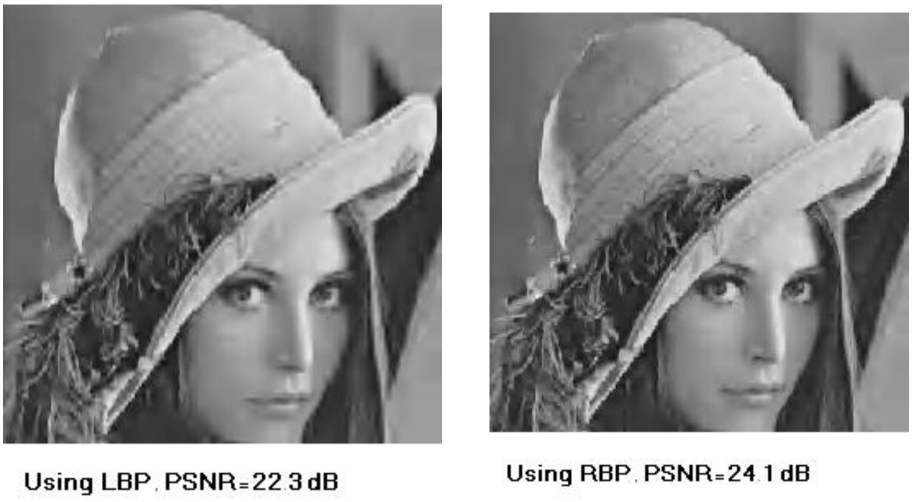
Fig. (4) Testing results for 128:1 compression ratio with Q= 8 neurons.



Using LBP, PSNR=17.2 dB

Using RBP, PSNR=19.7 dB

Fig. (5) Testing results for 85:1 compression ratio with Q= 16 neurons.



Using LBP, PSNR=22.3 dB

Using RBP, PSNR=24.1 dB

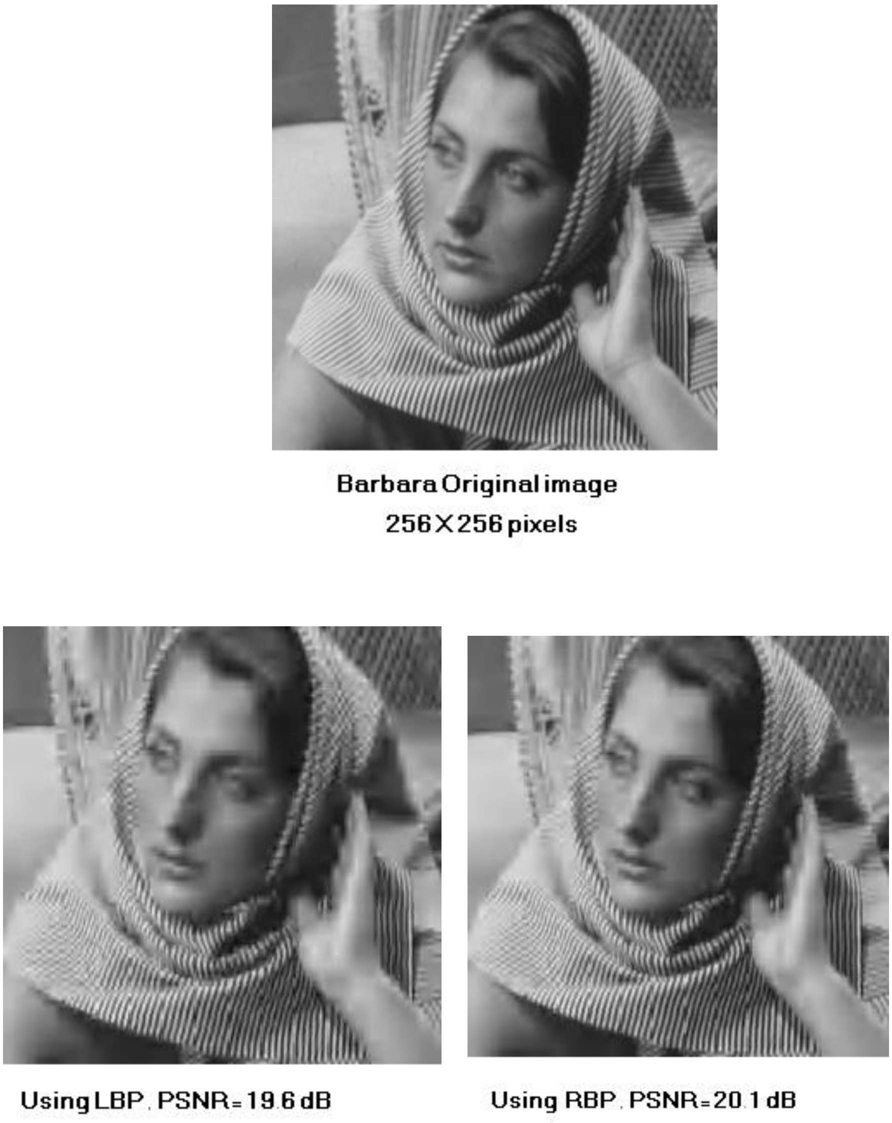
Fig. (6) Testing results for 64:1 compression ratio with Q= 16 neurons.

Tables (1 and 2) show the performance of RBP and LBP learning rules. The compression rate in the two tables reflected how accurately the images were decompressed. The two schemes showed high deterioration of the image quality for high compression rates and very little deterioration when using an appropriate number of neurons in the internal hidden layer (compressed layer) of the neural networks. As the number of neurons in the compressed layer was lowered, the greater deterioration in the decompressed images. However, the lower the number of compressed layer neurons, the better the compression ratio of the image.

As can be seen from figures (3 to 6) and figures (7 to 9 ,(which test image compression / decompression capabilities and performance of the RBP and LBP. All images suffered in quality with different degrees but no deterioration. A higher Peak Signal to noise ratio *PSNR* was achieved for RBP learning rule, and lower *PSNR* for LBP learning rule, for the same hierarchical neural network parameters. Overall, it was found that superior performance could be achieved with the proposed RBP based hierarchical neural network, while the LBP shows inferior performance.

6. Conclusions

This paper has successfully applied Resilient backpropagation RBP to a large, complex task. The results appear to be promising in image compression/ decompression problems. It is used to train a hierarchical neural network. We segmented, compressed, decompressed, and reconstructed various images using this method. A number of experiments have been conducted. Results showed that a superior PSNR could be achieved with proposed RBP compared with LBP.

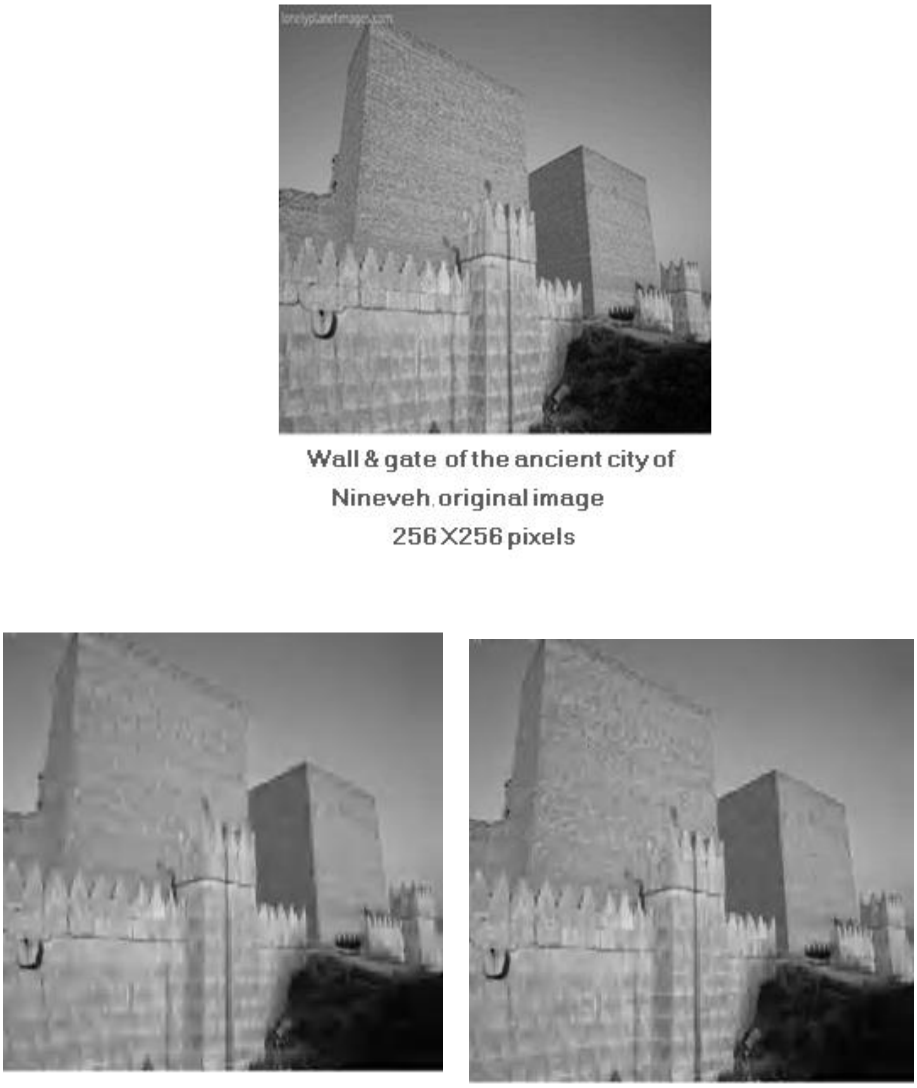


Barbara Original image
256X256 pixels

Using LBP, PSNR=19.6 dB

Using RBP, PSNR=20.1 dB

Fig. (7) Testing results for 64:1 compression ratio with Q= 16 neurons.

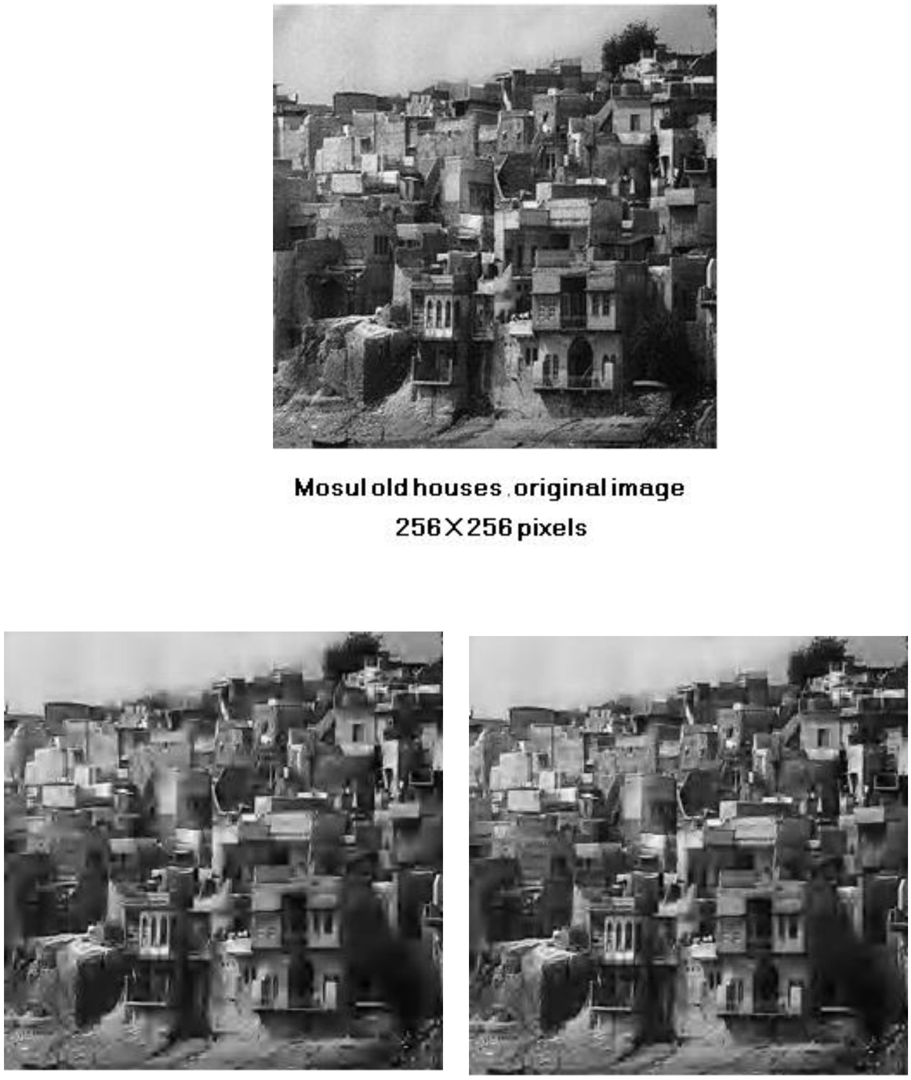


Wall & gate of the ancient city of
Nineveh, original image
256X256 pixels

Using LBP, PSNR=22.1 dB

Using RBP, PSNR=23.2 dB

Fig. (8) Testing results for 64:1 compression ratio with Q= 16 neurons.



Mosul old houses , original image
256X256 pixels

Using LBP , PSNR=20.1 dB

Using RBP , PSNR=22.5 dB

Fig. (9) Testing results for 64:1 compression ratio with Q= 16 neurons.

REFERENCES

[1] Amerijckx,C. ; J. Legat, and M. Verleysen, (2003) “ Image compression using self- organizing maps”, **System Analysis Modelling Simulation** ,Vol. 43, No. 11, pp. 1529-1543.

[2] Dony, R. D. and S. Haykin, (1995)"Neural network approaches to image compression", **Proc. IEEE** 83 (2), pp. 288- 303.

[3] Egmont-Ptersen, M.; D. de Ridder, and H. Handels, (2002)"Image processing with neural networks-a review", **Pattern Recognition**, Vol.35, pp.2279- 2301.

[4] E. Gelenbe, M. Sungur, C. Cramer, (2003) “Learning random networks for compression of still and moving images”, Dept. Ele. Engg., Duke University, http://www.ee.duke.edu./~cec/research/ neuralcom pression/JPL/paper.html}.

[5] Jiang, J. (1999) Image compression with neural networks – A survey”, Signal Processing Image Communication 14 pp.737–760.

[6] Jiang, J. (1996) “A neural network design for image compression and indexing”, Hawaii, USA, **International Conference on Artificial Intelligent Expert Systems and Neural Networks**, pp.296- 299.

[7] Jigrea, R. and L. Pajtek, (2004) “Neural networks image compression: a way to real time movie transmission over slow serial lines”, **Automation and Computer Science Faculty**, University of Timeworn, Romania, http://citeseer.ist.psu.edu/correct/975252.

[8] M. Riedmiller, H. Braun, (1993) “A direct adaptive method for faster backpropagation learning, The RPROP Algorithm”, **IEEE ICNN**, pp. 586-591.

[9] Rumelhart, D.; G. Hinton, and R. Williams, (1986) “Learning internal representations by error propagation”, **in Parallel Distributed Processing, MIT Press**, Vol. (1), pp. 318-362.

[10] Rafid A. Khalil, (2005) " Image compression with backpropagation neural networks", Submitted to **Journal of Prospective Researches**.

[11] Rizvi, S. and L. Wang, (1997) “Nonlinear vector prediction using feedforward neural network”, **IEEE Trans. Image Process**., 6(10), pp.1431-1436.