

Neural Network Technique for Lossless Image Compression Using X-Ray Images

N. Senthilkumaran, *Member IACSIT* and Dr. J. Suguna

Abstract—Neural Networks are based on the parallel architecture and inspired from human brains. Neural networks are a form of multiprocessor computer system, with simple processing elements, a high degree of interconnection, simple scalar messages and adaptive interaction between elements. One such application is image compression. Image compression is a process which minimizes the size of an image file without degrading the quality of the image to an unacceptable level. It also reduces the time required for images to be sent over the internet or downloaded from web pages. This paper proposes an Improved Backpropagation Neural Network Technique, for lossless image compression. The system also proves that the improved Backpropagation Neural Network Technique works better than the existing Huffman Coding Technique for lossless image compression by considering X-Ray images based on three metrics such as compression ratio, transmission time and compression performance. Experimental results are presented and compared.

Index Terms—Backpropagation, Huffman Coding, Image Compression, Neural Network, X-Ray.

I. INTRODUCTION

Images play an important role in the world of multimedia and its transmission with storage has become really a big burden as it occupy more space in memory[5][8]. The goal of image compression is to create smaller files that use less space to store and less time to send[3][9]. Image compression involves reducing the size of image data files, while retaining necessary information [1] [7] [13]. Hence, it is essential to analyze and suggest a best technique for lossless image compression [12][14].

The main objective of this paper is to compare the existing Huffman Coding Technique with the Improved Backpropagation Neural Network Technique for lossless image compression. The suggested technique should, reduce the time to send the images over the internet or downloaded from web pages, Reduce the memory space required to store the image and Compress the image without any loss of information. In order to achieve the above objectives, it is essential to improve the Backpropagation Neural Network Technique, compare it with the Huffman Coding Technique by considering different metrics and analyze the results obtained.

Manuscript received March 20, 2010.

N. Senthilkumaran is with the School of Computer Science and Engineering, Bharathiar University, Coimbatore - 641 046, India. (e-mail:senthilkumaran@ieee.org).

Dr. J. Suguna is with the Computer Science Department, Vellalar College for Women, Erode, India.

II. NEURAL NETWORKS

Neural networks are simplified models of the biological neuron system and therefore have drawn their motivation from the computing performed by a human brain [2] [6] [11]. A neural network, in general, is a highly interconnected network of a large number of processing elements called neurons in an architecture inspired by the brain[4][10]. Artificial neural networks are massively parallel adaptive networks of simple nonlinear computing elements called neurons which are intended to abstract and model some of the functionality of the human nervous system in an attempt to partially capture some of its computational strengths[13][14]. A neural network can be viewed as comprising eight components which are neurons, activation state vector, signal function, pattern of connectivity, activity aggregation rule, activation rule, learning rule and environment [10].

A. Artificial Neuron

The human brain no doubt is a highly complex structure viewed as a massive, highly interconnected network of simple processing elements called neurons. However, the behavior of a neuron can be captured by a simple model as shown in Figure 1. Every component of the model bears a direct analogy to the actual constituents of a biological neuron and hence is termed as artificial neuron [2] [7]. It is this model which forms the basis of Artificial Neural Networks.

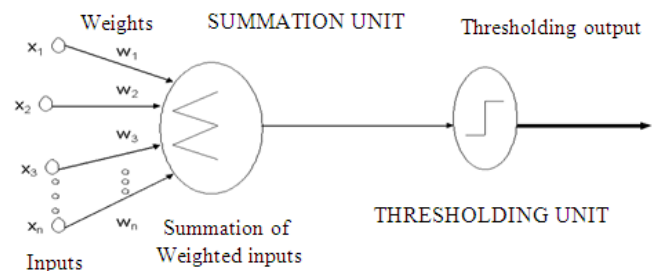


Figure 1 Simple model of an artificial neuron

Here, $x_1, x_2, x_3 \dots x_n$ are the n inputs to the artificial neuron, $w_1, w_2, w_3, \dots w_n$ are the weights attached to the input links. A biological neuron receives all inputs through the dendrites, sums them and produces an output if the sum is greater than a threshold value. The input signals are passed on to the cell body through the synapse which may accelerate or retard an arriving signal [3] [7] [11]. It is this acceleration or retardation of the input signals that is modeled by the weights. An effective synapse which transmits a stronger signal will have a correspondingly larger weight while a weak synapse will have smaller weights. Thus, weights here are multiplicative factors of the inputs to account for the synapse.

Hence, the total input (**I**) received by the soma of the artificial neuron is

$$I = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

$$I = \sum_{i=1}^n w_i x_i$$

B. Backpropagation Neural Network

Backpropagation is a systematic method of training Multilayer Artificial Neural Networks. The Backpropagation derives from the fact that computations are passed forward from the input to the output layer. The Feed forward Backpropagation Network is a very popular model in Neural Networks [6] [9] [10]. In Multilayer Feed forward Networks, the processing elements in adjacent layers are connected. This is represented by the following figure.2

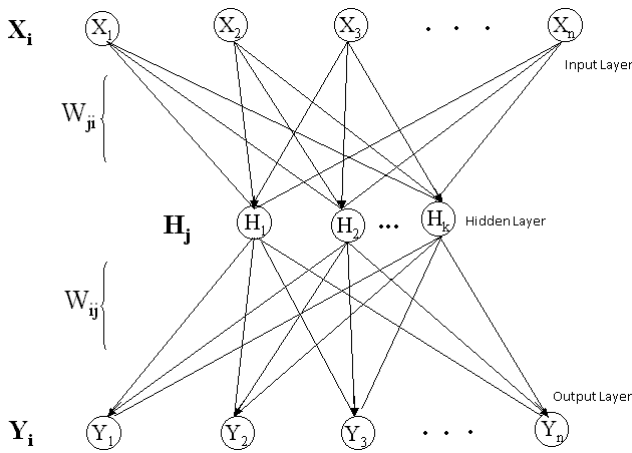


Figure.2 Multilayer feed forward Backpropagation Network

The Feed forward process involves presenting an input pattern to input layer neurons that pass the input values onto the hidden layer. The hidden layer nodes compute a weighted sum of its inputs and present the result to the output layer.

III. IMAGE COMPRESSION

Image Processing is a very interesting and a hot area where day-to-day improvement is quite inexplicable and has become an integral part of own lives. Image processing is the analysis, manipulation, storage, and display of graphical images. An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk or CD-ROM [1] [7] [10]. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer. Once the image has been digitized, it can be operated upon by various image processing operations. Image processing is a module that is primarily used to enhance the quality and appearance of black and white images. It also enhances the quality of the scanned or faxed document, by performing operations that remove imperfections. Image processing operations can be roughly divided into three major categories, Image Enhancement, Image Restoration and Image Compression.

Image compression is familiar to most people. It involves reducing the amount of memory needed to store a digital image. Images play an important role in the world of

multimedia and its transmission with storage has become really a big burden as it occupy more space in memory. So the concept of image compression evolved. By compressing the images, users find the storage media much more voluminous. So Image Compression enhances the progress of the world in communication [4] [7] [11].

A. Need for Image Compression

Images are stored on computers as collections of bits representing pixels, forming the picture elements. Many pixels are required to store even moderate quality images. Image compression plays a pivot role in diminishing this amount of information. Most images contain some amount of redundancy that can sometimes be removed when the image is stored and replaced when it is reconstructed, but eliminating this redundancy does not lead to high compression. The amount of data associated with visual information is so large that its storage would require enormous storage capacity. So image compression is very important to reduce the storage and transmission costs while maintaining good quality. Image compression is the process of effectively coding digital images to reduce the number of bits required in representing an image. If the compression is effective, the resulting stream of codes will be smaller than the original symbols[3][9][11].

B. Image Compression Methods

Image Compression has been pushed to the forefront of the image processing field. This is largely a result of the rapid growth in computer power, the corresponding growth in the multimedia market, and the advent of the World Wide Web, which makes the internet easily accessible for everyone. Compression algorithm development starts with applications to two-dimensional (2-D) still images. Because video and television signals consist of consecutive frames of 2-D image data, the development of compression methods for 2-D still data is paramount importance[1][3][6][9].

The goal of image compression is to create smaller files that use less space to store and less time to send. Image compression involves reducing the size of image data files, while retaining necessary information. The reduced file is called the compressed file and is used to reconstruct the image, resulting in the decompressed image. The original image, before any compression is performed, is called the uncompressed image file. The ratio of the original, uncompressed image file and the compressed file is referred to as the compression ratio. The compression ratio [5] is denoted by:

$$\text{Compression Ratio} = \frac{\text{Size of the output file}}{\text{Size of the input file}}$$

$$\text{Transmission Time} = \frac{\text{Number of pixels X Number of Bits/pixels}}{\text{Modem Speed (kilobits/ second)}}$$

$$\text{Compression performance} = 100 \times (1 - \text{Compression ratio})$$

The basic types of image compression methods are Lossless compression method and Lossy compression

method. Lossless compression is a compression without any loss of image quality. This means the compression method will not cause any loss of data or errors. This is possible because all read data contain repeating patterns of some sort that a compression processor can search out and then arrange to transmit more efficiently. Generally these methods are used in applications where the loss of even a single bit is dangerous [7] [9] [10].

In order to achieve high compression ratios with complex images, lossy compression methods are required. Lossy compression provides tradeoffs between image quality and degree of compression, which allows the compression algorithms to be customized to the application. With some of more advanced methods, images can be compressed 10 to 20 times with virtually no visible information loss, and 30 to 50 times with minimal degradation. Image enhancement and restoration techniques can be combined with Lossy compression schemes to improve the appearance of the decompressed image. In general, a higher compression ratio results in a poorer image, but the results are highly image dependent. A technique that works well for one application may not be suitable for another. Lossy compression is a compression with loss of image quality. In this compression method, the image is not an exact replacement of the original image [10] [14].

IV. PROPOSED TECHNIQUE FOR LOSS LESS IMAGE COMPRESSION

- Step 1:* Select the input image for compression.
- Step 2:* Compress the selected image using Huffman Coding Technique and Improved Backpropagation Neural Network Technique
- Step 3:* Decompress the compressed file using Huffman Coding Technique and Improved Backpropagation Neural Network Technique
- Step 4:* Save the compressed and decompressed files and Calculate the compression ratio, transmission time and Compression performance
- Step 5:* Analyze the results for the above metrics

The proposed technique for lossless image compression is shown in the following Figure. 3.

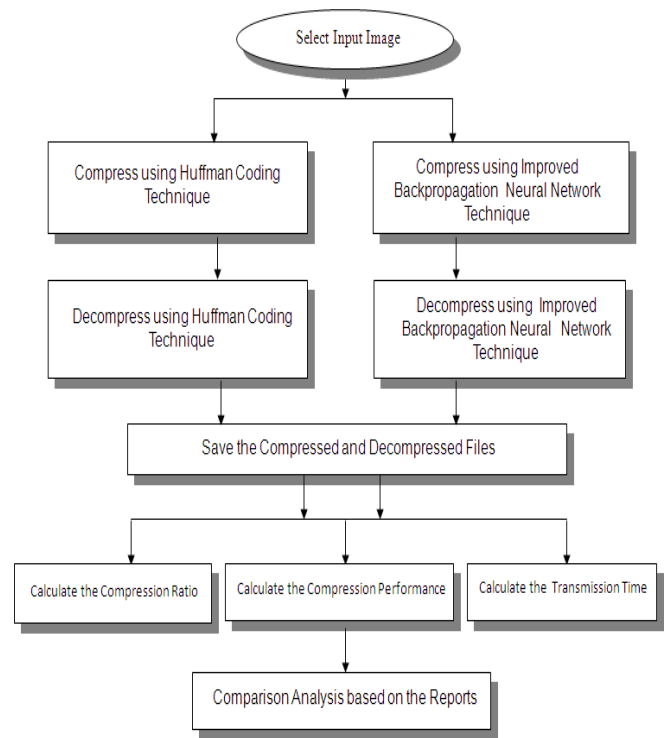


Figure. 3 Proposed Technique for lossless image compression

V. HUFFMAN CODING TECHNIQUE FOR LOSS LESS IMAGE COMPRESSION

A. Image Compression Process

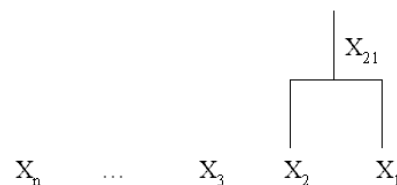
- Step 1:* Get the input image using 24-bit BMP format and convert the image file format into vector format (X)

$$X = (X_1 X_2 X_3 \dots X_n)$$
- Step 2:* Assign the index code (I) using pixel values of vector format (X).

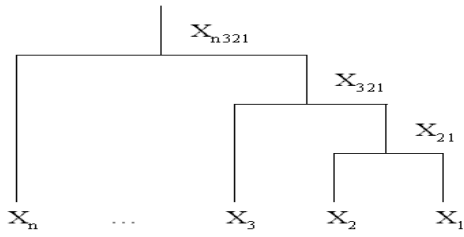
$$(X_1 X_2 X_3 \dots X_n)$$

$$(I_1 I_2 I_3 \dots I_n)$$
- Step 3:* Arrange the pixel values from highest to lowest order.

$$(X_n \dots X_3 X_2 X_1)$$
- Step 4:* Select the two lowest pixel values (X), grouped together and sum them. This begins the construction of a “Huffman Tree” structure.

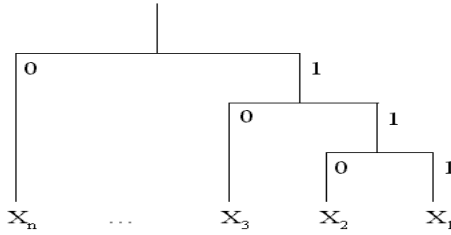


- Step 5:* Continue the same process until a Huffman Tree is completed.



The resultant tree obtained refers the Huffman Tree

Step 6: Each branch of the tree is labeled with either 0 or 1.



Step 7: Tracing down the tree gives the Huffman code.

$$H = (01011 \dots 111)$$

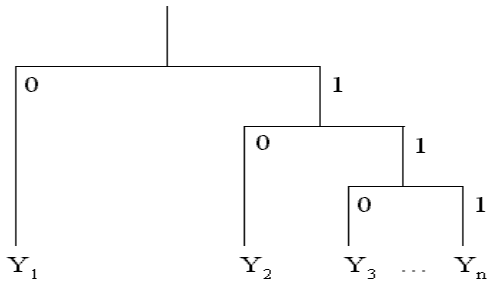
The resultant values (H) obtained refers the compressed file.

B. Image Decompression Process

Step 1: Get the Huffman code (H) as input from the compressed file.

$$H = (01010 \dots 111)$$

Step 2: The Huffman code (H) is assigned to each branch of the tree [4].



Step 3: Using Huffman Tree, assign the binary coded values to the Huffman decode values (Y).

$$Y = (Y_1 Y_2 Y_3 \dots Y_n)$$

Step 4: Index the Huffman decode values (Y), using index code (I).

$$(Y_1 Y_2 Y_3 \dots Y_n)$$

$$(I_n \dots I_3 I_2 I_1)$$

Step 5: Arrange the index code (I) in ascending order.

$$(I_1 I_2 I_3 \dots I_n)$$

Step 6: Arrange the Huffman decode values (Y) using the index code.

$$Y = (Y_n \dots Y_3 Y_2 Y_1)$$

Step 7: Now the Huffman decode values (Y) is converted into the image file format.

VI. IMPROVED BACK PROPAGATION NEURAL NETWORK TECHNIQUE FOR LOSS LESS IMAGE COMPRESSION

A. Image Compression Process

Step 1: Get the input image using 24-bits BMP format and convert the image file format into a matrix format (I).

$$I = \begin{pmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2n} \\ X_{31} & X_{32} & X_{33} & \dots & X_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & X_{m3} & \dots & X_{mn} \end{pmatrix}_{m \times n}$$

Step 2: Using the matrix format (I), count the pixel value and the number of occurrences of the neighboring pixel values. Representing these pair values (P) as follows.

$$P = (U_1, V_1) (U_2, V_2) (U_3, V_3) \dots (U_i, V_j)$$

Where, U = Pixel values.

V = Number of occurrences of the neighboring pixel values.

Step 3: The pair values (P) obtained from the above step can be represented in sequence order (S) [4].

$$S = U_1, V_1, U_2, V_2, U_3, V_3 \dots U_i, V_j$$

Step 4: The sequence order (S) can be provided as an input (X_i) to the Multi-Layer Feed-Forward Backpropagation Neural Network.

$$X_i = X_1, X_2, X_3 \dots X_n$$

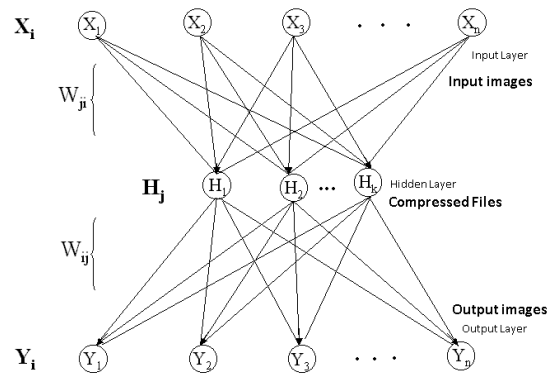


Figure. 4. Multi-Layer Feed-Forward Backpropagation Neural Network Architecture

Step 5: Calculate the weight (W_{ji}) using the formula [22].

$$W_{ji} = \sum_{i=1}^n X_i X_i^T$$

Where, $1 \leq j \leq k$,
 X_i is the input Layer.

Step 6: The Hidden Layer of the Multi-Layer Feed-Forward Backpropagation Neural Network can be created by using the formula (H_j) [2].

$$H_j = \sum_{i=1}^n W_{ij} X_i$$

Where, $1 \leq j \leq k$,
 X_i is input layer.

$H_j = H_1, H_2, H_3, \dots, H_k$

The result of the H_j obtained refers the compressed file.

B. Image Decompression Process

Step 1: Get the Hidden Layer (H_j) of the Multi-Layer Feed-Forward Back propagation Neural Network

$$H_j = H_1, H_2, H_3, \dots, H_k$$

Apply the Multi-Layer Feed-Forward Back-propagation Neural Network Architecture (Fig. 4).

Step 2: Calculate the weight (W_{ij}) using the formula [19].

$$W_{ij} = \sum_{j=1}^k H_j H_j^T$$

Where, $1 \leq i \leq n$,
 H_j is the Hidden Layer.

Step 3: The Output Layer of the Multi - Layer Feed - Forward Backpropagation Neural Network can be created by using the formula (Y_i) [1].

$$Y_i = \sum_{j=1}^k W_{ij} H_j$$

Where, $1 \leq i \leq n$,
 H_j is Hidden layer.

$Y_i = Y_1, Y_2, Y_3, \dots, Y_n$

Step 4: The resultant Output Layer (Y_i) of the Multi - Layer Feed-Forward Backpropagation Neural Network obtained from the above step can be represented in Sequence Order (S).

$$S = U_1, V_1, U_2, V_2, U_3, V_3 \dots U_i, V_j.$$

Step 5: The Sequence Order (S) Value can be represented in Pair Values (P). Each Pair represents the Pixel Value and the number of occurrences of the neighboring pixel values.

$$P = (U_1, V_1) (U_2, V_2) (U_3, V_3) \dots (U_i, V_j).$$

Where,
 U = Pixel values.
 V = Number of occurrences of the neighboring pixel values.

Step 6: All the Pair Values (P) represented in Pixel Values can be converted into Matrix Format (I).

$$I = \begin{pmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2n} \\ X_{31} & X_{32} & X_{33} & \dots & X_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ X_{m1} & X_{m2} & X_{m3} & \dots & X_{mn} \end{pmatrix}_{m \times n}$$

Step 7: Now the matrix format (I) is converted into the image file format.

VII. RESULTS AND DISCUSSION

This system uses three metrics such as compression ratio, transmission time and compression performance to compare and analyze the results with the results of Huffman Coding Technique. From the analysis, (Fig. 5,6,7,8,9) it is found that the system exhibits significant performance.

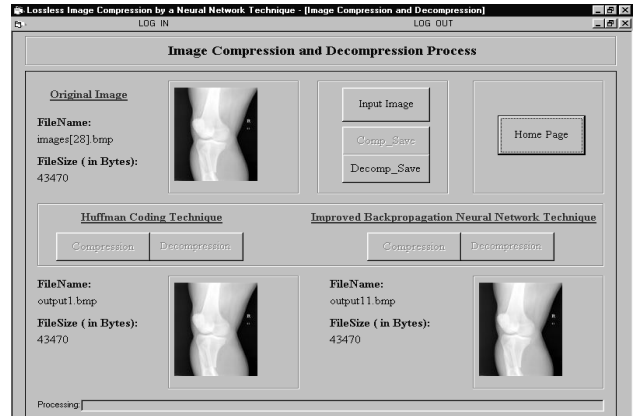


Figure 5. Lossless Image Compression using image1

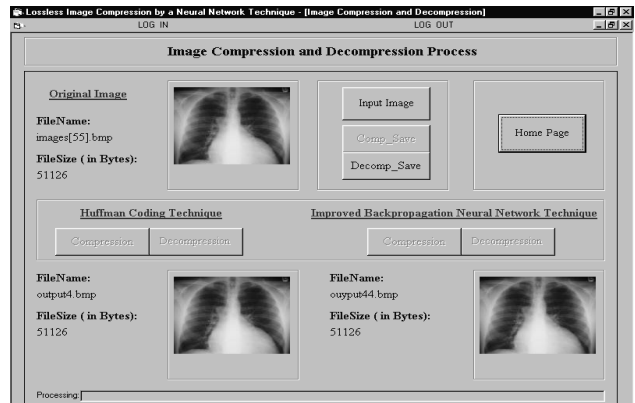


Figure 6. Lossless Image Compression using image2



Figure 7. Lossless Image Compression using image3

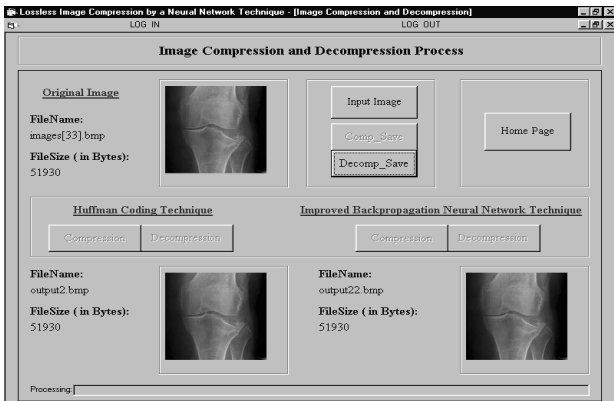


Figure 8. Lossless Image Compression using image4



Figure 9. Lossless Image Compression using image5

The following figures (Fig. 10,11,12) shows that the Improved Backpropagation Neural Network Technique is better than Huffman Coding Technique for lossless image compression.

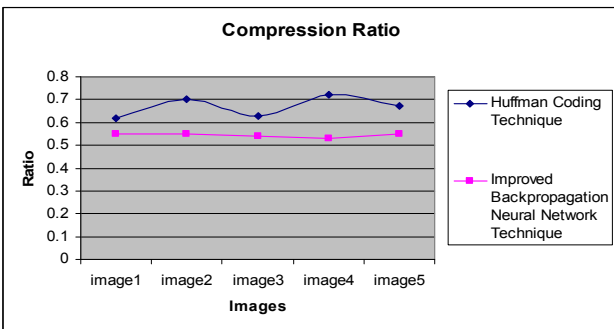


Figure 10. Compression Ratio Analysis

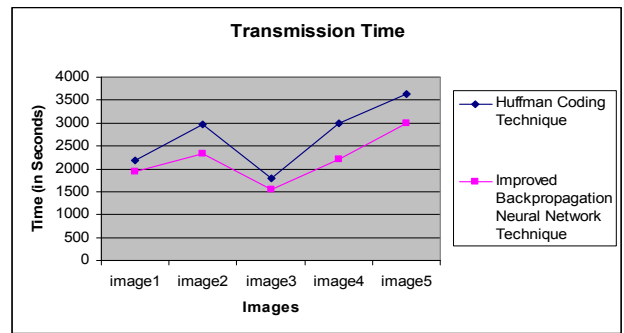


Figure 11. Transmission Time Analysis

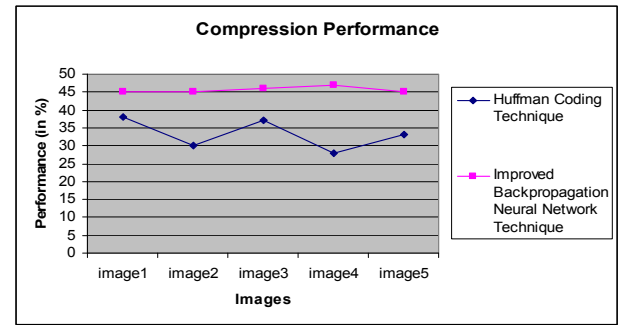


Figure 12. Compression Performance Analysis

VIII. CONCLUSION

In this paper, a Backpropagation Neural Network Technique is improved for lossless image compression. This system uses three metrics such as compression ratio, transmission time and compression performance to compare and analyze the results with the results of Huffman Coding Technique. From the analysis, it is found that the system exhibits significant performance. This shows that the Improved Backpropagation Neural Network Technique is better than Huffman Coding Technique for lossless image compression. Surely this technique will open a scope in the field of medicine where every bit of information is significant.

REFERENCES

- [1] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, pp. 4-22, April 1987.
- [2] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, no. 3, pp. 349-389, March 1981.
- [3] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459-473, 1989.
- [4] G. L. Sicuranzi, G. Ramponi, and S. Marsi, "Artificial neural network for image compression," *Electronics Letters*, vol. 26, no. 7, pp. 477-479, March 29 1990.
- [5] D. Anthony, E. Hines, D. Taylor, and J. Barham, "A study of data compression using neural networks and principal component analysis," in *Colloquium on Biomedical Applications of Digital Signal Processing*, 1989, pp. 1-5.
- [6] R. Kohno, M. Arai, and H. Imai, "Image compression using a neural network with learning capability of variable function of a neural unit," in *SPIE Vol. 1360 Visual Communications and Image Processing '90*, 1990, pp. 69-75.
- [7] L. E. Russo and E. C. Real, "Image compression using an outer product neural network," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing '92*, San Francisco, CA, March 23-26 1992, pp. II 377-380.

- [8] N. M. Nasrabadi and Y. Feng, "Vector quantization of image based upon a neural-network clustering algorithm," in *SPIE Vol. 1001 Visual Communications and Image Processing '88*, 1988, pp. 207–213.
- [9] M. Manohar and J. C. Tilton, "Compression of remotely sensed images using self organized feature maps," in *Neural Networks for Perception*, H. Wechsler, Ed. San Diego, CA: Academic Press, 1992, vol. 1: Human and Machine Perception, pp. 345–367.
- [10] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen, "Neural networks for vector quantization of speech and images," *IEEE J. on Selected Areas in Communications*, vol. 8, no. 8, pp. 1449–1457, October 1990.
- [11] Carato, S., (1992), *Neural Networks for Image Compression*, Neural Networks Advances and Applications 2, (Ed) Gelenbe E., Elsevier Science Publishers, pp 177-198.
- [12] Namphol, A. et al., (1996), *Image Compression with a Hierarchical Neural Network*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 32, No. 1, pp. 327-337.
- [13] Dony, R. D., and Haykin, S., (1995), *Neural Network Approaches to Image Compression*, Proceedings of the IEEE, Vol. 23, No. 2, pp 289-303.
- [14] Min, K., and Min, H. (1992) *Neural Network Based Image Compression using AMT DAP 610*, SPIE Vol. 1709, Application of Artificial Neural Networks, pp 386-393.



Mr. N. Senthilkumaran received his MCA degree from Bharathidasan University, India, and M.Phil. in Computer Science from Bharathiar University in 2002 and 2005 respectively.

He is currently doing his Ph.D. degree in Computer Science in the School of Computer Science and Engineering, Bharathiar University, India, under the Guidance of Dr. R. Rajesh. His research interests are Digital Image Processing, Soft Computing. He has for his credit 15 international

journal/conference papers. He is a student member of IEEE.



Dr. J. Suguna received the master's degree in mathematics from Annamalai University, Chidambaram in 1988 and the Ph.D. degree in computer science from the Bharathiar University, Coimbatore in 2009.

She is currently an Associate Professor with the Department of Computer Science, Vellalar College for Women (Autonomous), Erode, Tamil Nadu. Her research interests are AI, Data Mining, Text Mining and Image Processing. She is the author or co-author of over 20 publications in journals, conference

proceedings and book chapters. She has presented a paper in an International Conference held at Cincinnati University, Cincinnati, Ohio, USA. She has produced over 10 M.Phil. scholars in computer science. She is a board of studies member for computer science, Bharathiar University, Coimbatore and an annual member of the Indian Science Congress Association, Kolkata.