International Journal of Electrical and Computer Engineering.
ISSN 0974-2190 Volume 3, Number 2 (2011), pp. 83-94

http://www.irphouse.com

An Efficient Image Compression Technique using Artificial Neural Network with Huffman Coding

Vishal Awasthi[1] and Shri Ram Gupta[2]

[1]Member, IETE, Deptt. of Electronics Engg.,
CSJM University, Kanpur-24, U.P., India
[2]Department of Electronics Engg.,
CSJM University, Kanpur-24, U.P., India
E-mail: awasthiv@rediffmail.com [1]

Abstract

Image Compression using Artificial Neural Networks (ANN) is significantly different than compressing raw binary data. General purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. The classic image compression techniques have serious limitations at high compression rate. In this paper a feed forward Networks with Back propagation Algorithm adopting the method of steepest descent is used to minimize the performance error and a lossless coding technique i.e., Huffman coding to achieve better image compression ratio. The proposed method also includes Cumulative Distribution Function (CDF) to improve convergence time by breaking large images into smaller windows. After applying the proposed algorithm, results shows a great improvement in terms of compression ratio and peak signal to noise ratio (PSNR) than the performance of linear backpropagation schemes.

Keywords: Image Compression, Artificial Neural Network, Back-propogation Algorithm, Huffman Coding.

Introduction

Image compression is a much studied topic these days and is growing steadily. Large number of applications as remote sensing, tele-video conferencing and medical imaging involves huge data storage and transmissions. Employing large size memory devices involves highre cost and hence it is necessary to compress the data and preserve it using small memory devices. Usually image compression is applied prior to the storage or transmission of the image data. Later the compressed image is decompressed to get the original image. In general the reduction of image data is achieved by the removal of redundant data.

Data compression is a process of reducing the amount of data required to represent a given quantity of information. It minimizes the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space.

Compression may be defined as transforming the two-dimensional pixel array into a statistically uncorrelated data set and data redundancy is an important term used in digital image compression. If m_1 and m_2 denote the number of words used to represent the same information, then the relative data redundancy D_r of the first data set can be defined as-

$$D_r = 1 - \frac{1}{C_r} \tag{1}$$

Where C_r is called the compression ratio and is given by $C_r = {}^{m_1}/_{m_2}$

The many benefits of image compression include less required storage space, quicker sending and receiving of images, and less time lost on image viewing and loading. In certain industries such as health, the archiving of large numbers of images is required. Depending on the type of compression applied, images can be compressed to save storage space. Regardless of industry, image compression has virtually endless benefits wherever improved storage, viewing and transmission of images are required.

The transmission of an image from one machine to another can be very time consuming due to the requirement of large amount of memory for storage. By using image compression techniques, it is possible to remove some of the redundant information contained in images, requiring less storage space and less time to transmit [1]. Artificial neural networks ANNs have been applied to image compression problems, have demonstrated their superiority over traditional methods, when providing high compression rate, and high signal to noise ratio [2]. Several literatures were discussed the subject of applying ANNs to image compression in detail as in [1, 3].

Annadurai [4] used feed forward networks using Back propagation algorithm adopting the method of Steepest descent for error minimization and is directly applied to image compression. If the gray levels of the neighbors with the pixel is minimum, then compression ratio as well as the convergence of the network can be improved. Riedmiller and braun [5] in 1993 proposed a new learning algorithm, RPROP, for multilayer feedforward networks. To overcome the inherent disadvantages of pure gradient-descent, RPROP performs a local adaptation of the weight-updates according to the behavior of the errorfunction. This leads to an efficient and transparent adaptation process. In 1952, an optimum algorithm of coding an ensemble of messages consisting of a finite number of members is suggested by David A. Huffman [6] for construction of minimum redundancy code in such a way that the average number of coding digits per message is minimized. Some of the most used algorithms and advanced schemes for image compression are standard backpropagation [3], linear backpropagation [7] and backpropagation neural networks [8,9] respectively. M. Worrell [10] adopted the Vector quantisation (VQ) architecture that has a fast autonomous learning algorithm suitable for use in real time image compression, to increase the efficiency of the compression scheme depending on its the statistical properties. Neural networks because of their fast parallel search capabilities make good vector quantisers. The classic image compression techniques such as JPEG and MPEG have serious limitations at high compression rate. R.A. Khalid [11] proposed a bipolar sigmoidal backpropagation BBP algorithm to train a feedforward autoassociative neural network by breaking large images into smaller windows for image compression/ decompression processes.

In this paper, we applied the artificial neural network with huffman coding algorithm to improve Peak signal to noise ratio (PSNR) , compression ratio and convergence time for efficient image processing.

Techniques of Image Compression

The level of image compression achieved can be represented by compression ratio. The compression ratio is obtained by dividing the data size of the original image by the data size of the compressed image. For better the compression, the ratio should be higher. The image compression can be classified into two major categories : (a) Lossless compression (b) Lossy compression

In most of the cases, it is necessary to maximize the compression ratios still meeting the quantities such as time to compress, time to decompress, computational cost, and the quality. The image compression and decompression operations are said to be symmetrical operations.

Lossless Compression

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely without error. It preserves the exact data content of the original image.

Lossless compression is possible because most real-world data has statistical redundancy. The Lempel–Ziv (LZ) compression methods are among the most popular algorithms for lossless storage. DEFLATE is a variation on LZ which is optimized for decompression speed and compression ratio, therefore compression can be slow. The very best modern lossless compressors use probabilistic models, such as prediction by partial matching. The Burrows–Wheeler transform can also be viewed as an indirect form of statistical modelling. In lossless image compression there is an intrinsic limitation as to how much image can be compressed. In general, the compression ratio achieved will be of the order of 3:1 for lossless approach.

In further refinement of these techniques, statistical predictions can be coupled with arithmetic coding, invented by Jorma Rissanen, and turned into a practical method to achieve superior compression. This resultant algorithm is well known as Huffman algorithm and lends itself especially well where the predictions are strongly context-dependent.

Lossy Compression

Another kind of compression, called lossy data compression or perceptual coding, is possible if some loss of fidelity is acceptable. Lossy image compression is used in digital cameras, to increase storage capacities with minimal degradation of picture quality. In lossy audio compression, methods of psychoacoustics are used to remove non-audible (or less audible) components of the signal. Compression of human speech is often performed with even more specialized techniques, so that "speech compression" or "voice coding" is sometimes distinguished as a separate discipline from "audio compression".

The idea of data compression is deeply connected with statistical inference. Lossy data compression systems typically include even more stages, including, for example, prediction, frequency transformation, and quantization. The entropy of an image is a measure of this limit. If the entropy is high, the image has little redundancy. If the entropy is low, the image contains high redundancy.

Problem Specifications

In general the images used for compression are of different types like dark image, high intensity image etc. When these images are compressed using Back-propagation Network, it takes longer time to converge. The reason for this is, the given image may contain a number of distinct gray levels with narrow difference with their neighborhood pixels. If the gray levels of the pixels in an image and their neighbors are mapped in such a way that the difference in the gray levels of the neighbors with the pixel is minimum, then compression ratio as well as the convergence of the network can be improved. To achieve this, a Cumulative distribution function is estimated for the image and it is used to map the image pixels. When the mapped image pixels are used, the Back-propagation Neural Network yields high compression ratio as well as it converges quickly. In order to achieve even better compression ratio, a lossless coding technique i.e., Huffman coding is used later on.

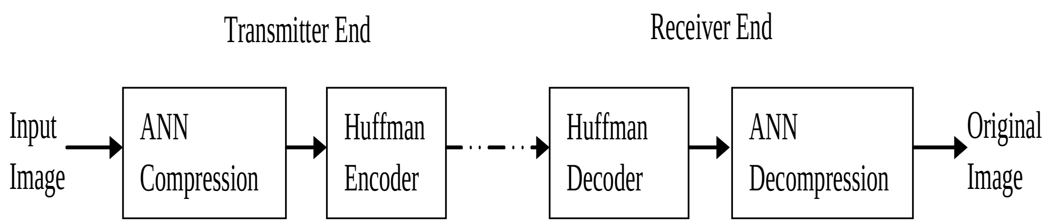

Figure 1: Block diagram of compression & decompression using ANN and Huffman

Artificial Neural Network

Artificial Neural Networks have been applied to image compression problems, due to their superiority over traditional methods when dealing with noisy or incomplete data. An Artificial Neural Network (ANN) is composed of simple elements operating in parallel. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between input elements so that a particular input leads to a specific target output.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Adaptive learning, Self-organisation, Real Time Operation and Fault Tolerance are some main advantages provided by neural network.

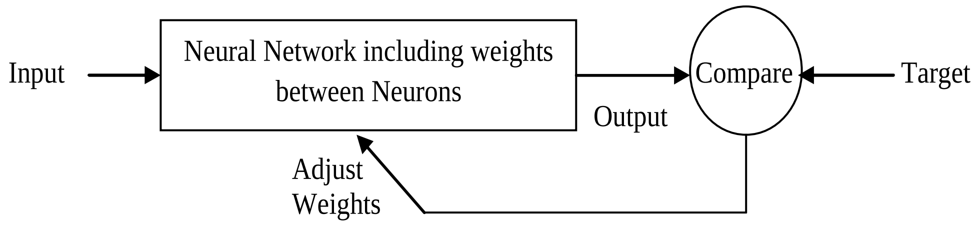

Figure 2: Block diagram of Neural Tetwork Tranning Algorithm.

A more sophisticated neuron is the McCulloch and Pitts model (MCP) in which the inputs are weighted i.e. the effect that each input has at decision making dependent on the weight of the particular input. These weighted inputs are added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

The Back-Propagation Algorithm

Artificial Neural Networks based techniques provide sufficient compression rates of the encoded data that is sent along a communication line and does not resemble its original form. These networks contain at least one hidden layer, with fewer units of the input and output layers. The Back Propagation Neural Network Algorithm performs a gradient-descent in the parameter space minimizing an appropriate error function along with the updation of weights.

Backpropagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Properly trained backpropagation networks tend to give reasonable answers when presented with inputs. This generalization property makes it possible to train a network on a representative set of input/target pairs and get optimal results.

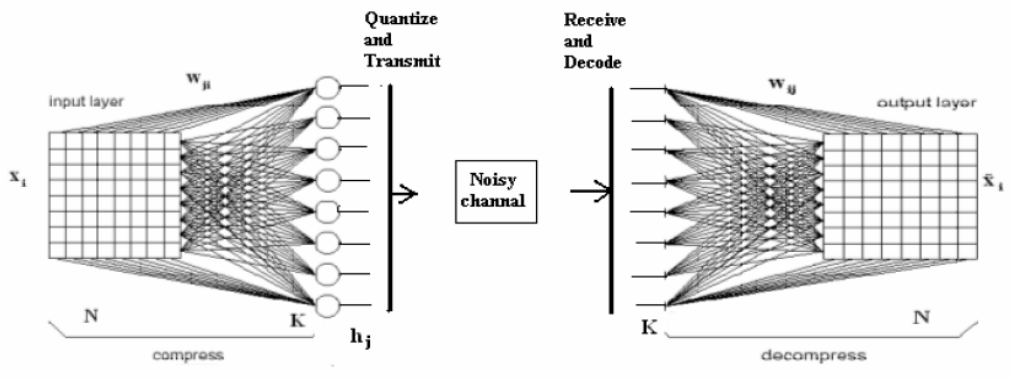

Figure 3: A Neural network Compression/ Decompression Pair.

Standard backpropagation is a gradient descent algorithm, in which the network weights are moved along the negative of the gradient of the performance function. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function.

Cumulative Distribution Function (CDF) and Huffman Technique

The Back-propagation Neural Network takes longer time to converge when used for compression of various types of images namely standard test images, natural images, medical images, satellite images etc. The compression ratio achieved is also not high. To overcome these drawbacks a new approach using Cumulative Distribution Function is proposed.

Mapping of Pixels by Estimating CDF of an Image

Computational complexity is involved in compression of raw pixels of an image in spatial domain or the mathematically transformed coefficients in frequency domain using Artificial Neural Networks. An image may contain a number of distinct gray levels with narrow difference with their neighborhood pixels. If the gray levels of the pixels in an image and their neighbors are mapped in such a way that the difference in the gray levels of the neighbor with the pixel is minimum, then compression ratio as well as the convergence of the network can be improved. To achieve this, the Cumulative Distribution Function is estimated for the image and it is used to map the image pixels. When the mapped image pixels are used, the Backpropagation Neural Network yields high compression ratio as well as it converges quickly.

The histogram of an image can be equalized by mapping the pixels through their cumulative distribution function $F_x(i)$. Typically, the pixels take on the values i= 0, …,L-1 where L is the number of discrete levels that a pixel can take on. The cumulative distribution function can then be approximated by-

$$F_x(i) = \frac{1}{h(L-1)} \sum_{j=0}^{j=1} h(j) \tag{2}$$

Histogram equalization does not introduce new intensities in the image. Existing values will be mapped to new values resulting image with less number of the original number of intensities. Mapping of the pixels by estimating the cumulative Distribution function of the image results in correlation of the pixels and the presence of similar pixel values within the small blocks of image augments the convergence of the network. Most of the image blocks will be similar and hence the learning time gets reduced. Since the convergence is quick, it is possible to reduce the number of neurons in the hidden layer to the minimum possible thus achieving high compression ratios without loss in quality of the decompressed image.

Huffman Technique for Compression

Huffman coding is an entropy encoding algorithm used for lossless data compression. In this coding technique, the variable-length code table has been derived based on the estimated probability of occurrence for each possible value of the source symbol.

Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code that expresses the most common source symbols using shorter strings of bits. The mapping of individual source symbols to unique strings of bits will produce a smaller average output size. For a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding, e.g. ASCII coding.

Huffman's algorithm is optimal for a symbol-by-symbol coding with a known input probability distribution. However, Huffman coding can be used in adaptively, accommodating unknown, changing, or context-dependent probabilities. A Huffman tree omits unused symbols and produces the most optimal code lengths.

Compression Procedure

This paper includes two levels of compression :- (i) Compression using artificial neural network and (ii) Compression using Huffman encoding. The $M \times N$ bit image is split into 4 x 4 pixels non-overlapping sub-images. The normalized pixel value of the sub-image will be the input to the nodes. The three-layered back propagation-learning network will train each sub-image. The number of neurons in the hidden layer will be designed for the desired compression.

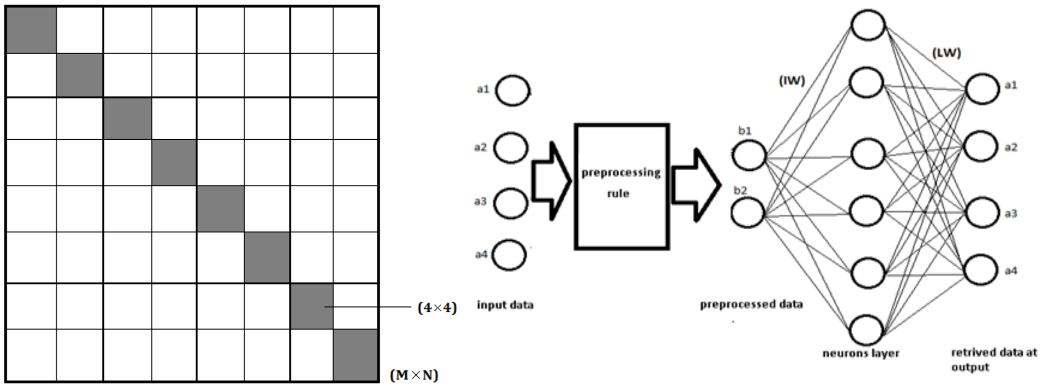

Figure 4: (a) Pixel representation of a block (b) Neural network with preprocessed data

The input image is split up into a number of blocks, each block has *N* pixels, which is equal to the number of input neurons see Figure 4. Initially the preprocessed data is obtained using mathematical rules. Now for training the samples of this compressed (preprocessed) data is fed to an artificial neural network which is trained for decompression. Thus the network provides us the necessary IW and LW matrices required for decompression of the image at receiver end. Now this preprocessed image data is fed to huffman encoder. After huffman encoding, the compressed image data file is obtained which is to be transmitted.

At receiver side, image data file and text file both are fed to huffman decoder. Thus huffman decoded data is obtained. Using this huffman decoded data and network parameters from the text file the original data image is retreived by realising the transfer function and after arranging this image data, the original image is restored.

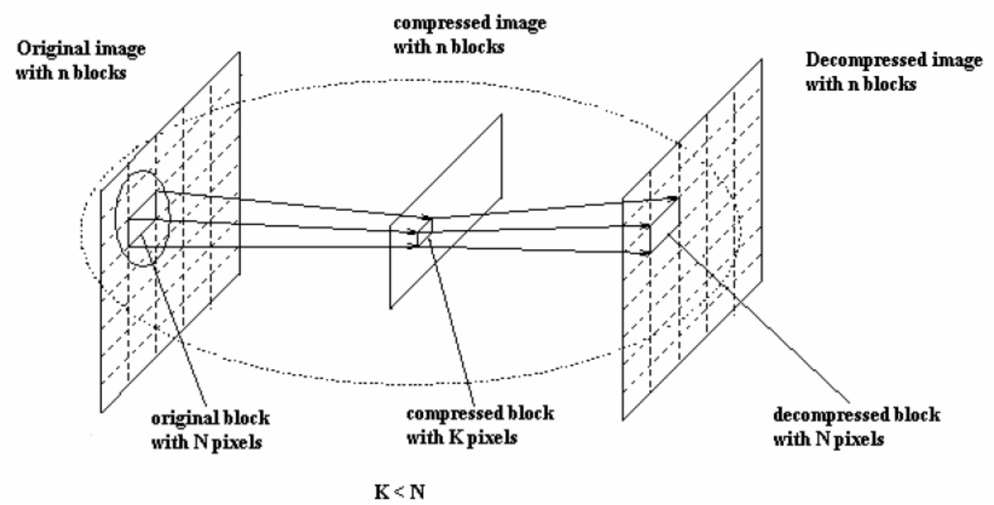

Figure 5: Compression of large image using a Neural Compressor/Decompressor

The range of the intensity level in the preprocessed data is reduced by range compression before Huffman coding. So that the total number of intensity levels in the data fed to Huffman encoder is reduced, resulting in smaller size Huffman codes and hence more compression.

Simulation Results and Discussion

The performance of the proposed approach has been tested in various types of images. The Artificial Neual Network was also tested by varying the number of neurons in the output layer with compression ratios of 25%, 50% and 75%. Table I and Table II listed below shows the data of Peak signal to noise ratio (PSNR) , compression ratio and convergence time with ANN and ANN with CDF & Huffman coding alogorithm respectively.

Table I: Compression with Mapping of Pixels.

For 25% ANN Compression				
S.No.	Image	CR	PSNR (db)	Time (sec.)
1.	Leena.bmp	0.2539	30.1959	2.3469
2.	Rice.png	0.2656	25.9880	2.5629
For 50% ANN Compression				
1.	Leena.bmp	0.5039	32.4059	13.9844
2.	Rice.png	0.5156	32.0808	10.7161
For 75% ANN Compression				
1.	Leena.bmp	0.7539	32.2191	20.1393
2.	Rice.png	0.7656	33.7513	19.1249

Table II (a): Compression without Mapping of Pixels.

For 50% ANN Compression				
S.No.	Image	CR	PSNR (db)	Time (sec.)
1.	Leena.bmp	0.5039	28.5015	14.6669
2.	Rice.png	0.5156	24.0431	13.0539

Table II (b): Compression Using ANN and Huffman Coding.

For 50% ANN Compression				
S.No.	Image	CR	PSNR (db)	Time (sec.)
1.	Leena.bmp	0.2775	30.8473	156.5647
2.	Rice.png	0.2726	42.1102	132.3546

Figure 6: Leena Image with compression ratio 50%. (a) Original image (b) Mapped by CDF(c) Decompressed image

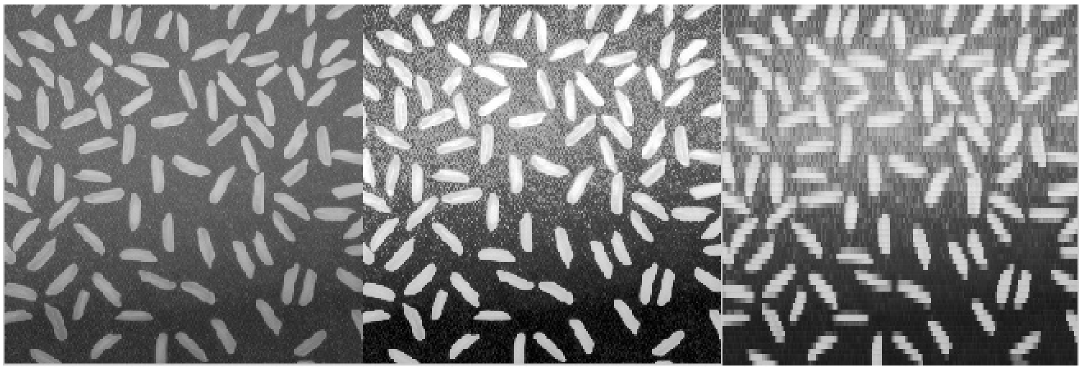

Figure 7: Rice Image with compression ratio 50%. (a) Original image (b) Mapped by CDF (c) Decompressed image

From the table II(a) i.e. Leena image for 50% compression, it is observed that, without mapping, the PSNR value is 28.5015db with convergence time 14.6669 seconds and compression ratio of 0.5039 and after using the ANN algorithm (table I), the PSNR value and conversion time is improved by 32.4059 db and 13.9844 seconds respectively but by applying the proposed approach i.e. ANN with huffman coding, the conversion time becomes 156.5647 seconds with a great improvement in PSNR and compression ratio value by amount of 30.4873 and 0.2775 respectively.

Similarly for Rice image for 50% compression, it is observed that, without mapping, the PSNR value is 24.0431db with convergence time 13.0539 seconds and compression ratio of 0.5156 and after using the ANN algorithm (table I), the PSNR value and conversion time is improved by 32.0808 db and 10.7161 seconds respectively but by applying the proposed approach i.e. ANN with huffman coding, the conversion time becomes 132.3546 seconds with a great improvement in PSNR and compression ratio value by amount of 42.1102 and 0.2726 respectively.

As can be seen from Tables I & II , the three schemes performed quite well. The compression rate in the three tables reflected how accurately the images were decompressed. The three schemes showed very little deterioration of the image quality when using an appropriate number of neurons in the hidden layer of the neural networks. As the number of hidden neurons was lowered, the greater deterioration in the decompressed images. The images results from testing compression / decompression capabilities and performance of there backpropagation schemes are shown in Figure 6 & 7. A higher peak signal to noise ratio (PSNR) was achieved for Back Propagation scheme. Mapping the image by Cummulative Distribution Function has helped the Back- Propagation Neural Network to converge easily compared to previous Techniques.

Conclusion

This paper has successfully applied back-propagation and huffman coding techniques in image compression/ decompression problems. It is used to train a feedforward autoassociative network. We segmented, compressed, decompressed, and reconstructed images using this method. The proposed approach of mapping the pixels by estimating the Cumulative Distribution Function is a simple method of pre-processing any type of image. There will not be any loss in data in the preprocessing and hence the finer details in the image are preserved in the reconstructed image. After the compression using neural network, if compressed image is encoded using huffman encoding, a better compression ratio can be achieved without making any degradation in image as the huffman encoding is a lossless compression technique. The quality of the decompressed image and the convergence time has been experimentally proved to be better than achieved by conventional methods and by the same algorithm without mapping the image by Cumulative Distribution function.

References

[1] M. Egmont-Ptersen, D. de Ridder, H. Handels, "Image processing with neural networks- a review", Pattern Recognition , 35, pp. 2279- 2301, 2002.

[2] B. Verma, M. Blumenstein, S. Kulkarni, "A new compression technique using an artificial neural network". Faculty of Information and Communication Technology, Griffith University, Australia, 2004.

[3] J. Jiang, "Image compression with neural networks – A survey, Signal Processing": Image Communication, 14, pp. 737 – 760, 1999.

[4] S. Anna Durai and E. Anna Saro “Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function” World Academy of Science, Engineering and Technology 17, 2006.

[5] *Martin Riedmiller and Heinrich Braun* “A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm” Institut fur Logik, Komplexitat und Deduktionssyteme, University of Karlsruhe, W-7500 Karlsruhe, FRG, 1993.

[6] *David A. Huffman "A Method for the Construction of Minimum-Redundancy Codes", Associate, IRE, 1952.*

[7] E. Gelenbe, M. Sungur, C. Cramer, "Learning random networks for compression of still and moving images", Dept. Ele. Engg., Duke University, 2003.

[8] Md. I. Bhuiyan, Md. K. Hassan, "Image compression with neural network using DC algorithm", Journal of signal processing, vol. 5, no. 6, pp. 445 -459, 2001

[9] A. K. Ashikur, C. Mofizur, "A new approach for compressing images using neural network", CIMCA Proceeding, Vienna – Austria, Feb. 2003.

[10] Worrell M., "Neural Networks for Image Processing Applications, IEE Colloquium" Advanced Processor Design, London, UK, pp.8/1 – 8/6, Oct.1992.

[11] Khalil R.A. "Digital Image Compression Enhancement Using Bipolar Backpropagation Neural Networks" Al-Rafidain Engineering Vol.15, No.4, pp. 1-13, Jan 2007.