ISSN 2250-3153

Image Compression Enhancement using Bipolar Coding with LM Algorithm in Artificial Neural Network

Prachi Tripathi

Electronics and Communication Engineering Division, Kanpur Institute of Technology, Gautam Budh Technical University (Formerly known as Uttar Pradesh Technical University), Lucknow, Uttar Pradesh, India

Abstract- The objective of image compression is to reduce irrelevance image data in order to store the image in less memory space and to improve the transfer time of the image. Without compression, file size is significantly larger, usually several megabytes, but with compression it is possible to reduce file size to 10 percent from the original without noticeable loss in quality. Even though there are so many compressions technique already presents a better technique which is faster and memory efficient. In this paper the Lossless method of Image Compression using Bipolar Coding Technique with LM algorithm in Artificial Neural Network is proposed by the author. The proposed technique is efficient, simple and suitable in implementation and requires less memory space. An algorithm based on the proposed technique has been developed and implemented in MATLAB platform to compress the input image.

Index Terms- Lossy and Lossless Image Compression, Artificial Neural Network, Bilopar Backpropagation coding Technique, Levenberg-Marquardt (LM) Algorithm, Peak Signal to Noise Ratio, PCA Technique, Error vs. Epochs Graphs.

I. INTRODUCTION

Image compression continues to be an important subject in many areas such as communication, data storage, computation etc. The existing traditional techniques are based on reducing redundancies in coding, inter-pixel and psycho visual representation [1]. The Performance of any image compression depends on its ability to capture characteristic features of the image, such as sharp edges and fine textures, while reducing the number of parameters used for its modeling [2].

Image compression requires in several aspects of real life applications such as satellite image data for weather forecast, earth resource applications, X-ray images, image communication and image database. The image(s) having pixel values between 0 to 255 and type gray scale are compressed. There are several techniques for the image compression but the major known techniques are Lossless image compression and Lossy image compression (refer section II of this paper). In this paper the author will use a lossless method of image compression and decompression with the help of bipolar coding technique and artificial neural network with feed forward propagation [3]. This technique is implemented and the better result obtained. In addition to this LM algorithm is also implemented for image compression and it is analyzed that bipolar coding with LM algorithm in ANN serve as a better and suitable technique for image compression [4].

The outline of the paper is as follows. The types of image compression are explained in Section 2. Section 3 provides the information regarding the artificial feed forward neural network. Section 4 explained the bipolar back propagation coding technique. The Levenberg-Marquardt algorithm is explained in Section 5. In Section 6, Performance parameters, results and discussion are presented in Section 7. Finally, section 8 presents the conclusion and future work.

II. TYPES OF IMAGE COMPRESSION

A. Lossy Image Compression

It is a data encoding method that compresses data by losing some of it. The process aim is to minimize the amount of data that needs to be held and transmitted by a computer. Lossy image compression is most commonly used to compress multimedia data such as audio, video and still images, especially in applications such as streaming media and internet telephony. Ideally lossy compression is transparent or imperceptible, which can be verified through an ABX test [6].

B. Lossless Image Compression

This is a class of data compression algorithms that allows the exact accurate original data to be reconstructed from the compressed data. The lossless compression is used in cases where it is important that the original and the decompressed data be identical, such as X-ray images. Some other examples are executable programs, text documents, and spreadsheet and source code. Some image having file formats of PNG or GIF, use only lossless compression [6].

III. ARTIFICIAL NEURAL NETWORK

An Artificial Neural Network (ANN) is a parallel information processing system which is better than linear computing method. ANN is mathematical model which is inspired by the human being nervous system, such as brain process system. ANN consists of a hundreds or thousands of interconnected artificial neurons to solve specific problems. An artificial neuron consists of Inputs, weights and mathematical activation function that determines the activation of a neuron. Another function computes the output of the neurons. Computation of the neurons depends upon the weight of the neurons. An ANN is applicable on data classification and pattern recognition through a learning process.

Artificial neural network is the clustering of the artificial neurons, occurs by creating interconnected layers which may be

vary as shown in figure [1]. Mostly all ANN have similar structure of topology.

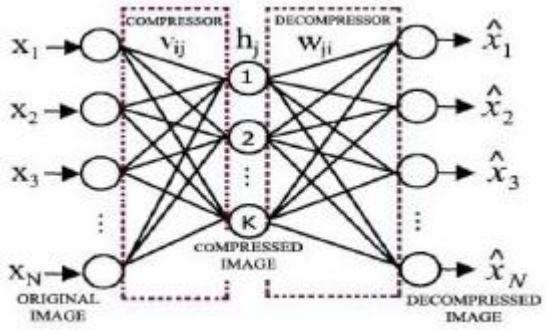

Figure 1: Basic compression structure

The input layer of each ANN contains the neurons that receive input from the outside environment and the output layer contains the neurons that communicate with the user of the environment. There are number of hidden layers between the input and output layers as shown in figure (2.2) below:

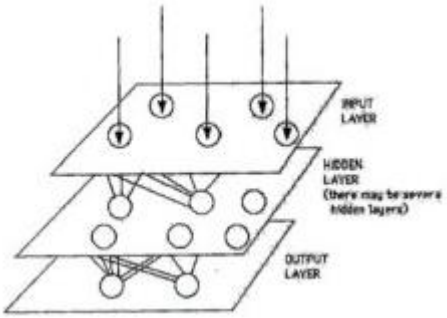

Figure 2: THREE peceptron for image compression

When the input layer neurons receives the inputs for the external environment, becomes the output for the other layer of the system. This output becomes the input for the third layer of ANN and this process goes on until a certain condition is satisfied or the layer fires the output to the environment. To determine the number of neurons in the network should have to perform trial and error method. If the specified input is linear scaling then there is no hidden neurons required. If the input is complex (non-linear scaling) the hidden number of neurons are increased too much an over fit occurs.

ANN is also called machine-learning algorithm because change in the weight of the artificial neurons causes the network to learn the solution of the problem. The strength of interconnected neurons is determined as a weight-value of the specific connection and the system learns new knowledge by adjusting these connection weights. The learning process of artificial neural network depends on its architecture and the algorithmic methods choose for the training. The training process is usually the combination of learning paradigms, learning rules and learning algorithms. There are three types of learning paradigm: supervised Learning, Unsupervised Learning and Hybrid learning. There are four basic types of learning rules: Error correction rules, Boltzmann rules, Hebbian and competitive learning.

The multilayer feed forward ANN usually applies the back propagation method for the training of network. In this method artificial neurons are organized in layers of ANN and send their signals "forward" and errors are propagated backwards. The backpropgation method applies supervised learning, means we want the network to compute and the error is calculated.

IV. Bipolar Back Propagation Coding Technique

The proposed Bipolar Coding technique using feed forward back propagation neural network converts decimal values into its equivalent binary code and reconvert in decompression phase. The compression and quality of image depends on number of neurons in the hidden layer. The quality of output image improves and the data loss reduces as the number of hidden neurons increases. The Bipolar Coding transformation is based on the bipolar activation function. The authors have proposed and applied this technique along with neural network for image compression. The sets of data obtained from an image are in analog form and required to convert into digital form. This conversion is possible using bipolar coding with linear scaling.

The conversions are based on certain ranges where analog form is scaled between value 0 and 1. Thus, for converting analog values into digital form, different binary values can be assigned. In this technique, each value is converted into the range between 0 and 1 using the formula as follows [1-2]

£ = X max - X min ……[4.1]

Y = Intercept C =(X - X min) / £…[4.2]

Now we convert the decimal values obtained from the above equations to binary ones as shown in figure[3]. When the value is between –0.1 and –0.75, then the code 000 is transmitted and when the value is between 0.5 and 0.75, the code 110 is transmitted.

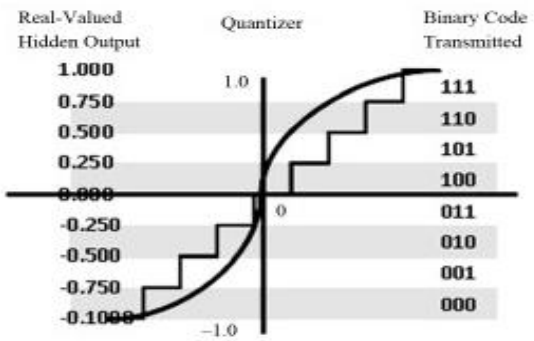

Figure 3: Digitization of hidden unit output

The output values of the neural network are used for calculation of the updated weights and biases in the next iteration. This problem of neural network learning can be considered as a function of optimization, where network is trying to determine the parameters like weights and biases to minimize the network errors.

The proposed Bipolar Coding technique using feed forward back propagation neural network is summarized in the following steps:

Step 1: Divide the image into small 8 × 8 chucks. These chucks are in square matrix form. It is easy to perform operation on such symmetric matrix.
Step 2: Obtain the values of pixels (0 to 255) of matrix and convert these values in the bipolar range –1 to 1. This is called as pixel to real number mapping.
Step 3: Now apply these values of bipolar range to the feed forward back propagation neural network. This neural network must have 64 input neurons as the chucks are of the size 8 × 8. Train the neural network using training algorithm as explained in Section 4.
Step 4: The bipolar values with weight and biases feed to the hidden layer which may have 2, 4, 8, 16, 32, and 64 hidden neurons. Convert these values in digital bits as explain in Section 5.
Step 5: Select the 8 × 8 chucks in a sequence after the completion of training of the neural network.
Step 6: Digital bits are now converted to real values.
Step 7: Matrix ranges bipolar values from –1 to 1 reconverted from real value to (0 to 255) pixel mapping.
Step 8: Finally, recall phase demonstrate the decompressed image (output image) at the output of the output layer of neural network. Pixel to real value and real value to pixel conversion is done during the compression and decompression process.

V. LEVENBERG-MARQUARDT ALGORITHM

A second tier feed-forward artificial neural network and the Levenberg-Marquardt algorithm are considered. Image encoding consists of the following steps:

Suppose, an image, F, is divided into mxl blocks of pixels. Every block is then scanned to define a input vector x (n) of size

p= m*l

It is assumed that the hidden layers of artificial neural network consists of L neurons each with P synapses(input) and it is characterized by an selected weight matrix Wi. All N blocks of the original image are passed through the hidden layer of ANN to obtain the hidden signals, h(n),which represent the encoded input image blocks, x(n). If L<P ,such coding gives image compression. It is assumed that the output layer of ANN consists of k = p = m x l neurons, each with L synapses(input). Let Wj be a selected output weight matrix. All N hidden signals h(n), representing an encoded image R, are processed through the output layer to obtain the output signal, y(n). The output signals are converted and reassembled into p (p = r x c) image blocks to obtain a reconstructed image, I'. There are two error matrices for the comparison of the quality of the image generated from the various image compression techniques: Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE). The MSE is the cumulative squared error between the compressed Images I' and the original image I.
MSE is defined as:
MSE = □ I

$$[\,I(x,y) - I'(x,y)\,]MN \sum\sum \text{.....[5.1]}$$

The quality of image encoding is defined by Peak-Signal-to-Noise ratio. PSNR is defined as:

PSNR = 20 log 10 [255/sqrt (MSE)] ……..[5.2]

Basic Algorithm:

Consider the Newton's method where the performance index is sum of squares. The Newton's method for optimizing a performance index F(x) is defined as

$X_{k+1} = X_k - A_k^{-1} G_k$ ……..[5.3]

Where $Ak = \nabla^2 F(x)$ and $G_k = \nabla F(x)$;

Supposed that F(x) is a sum of squares function:

$$F(x) = \sum_{r=1}^{n} v_i^2(x) = V^T(x)v(x)$$ …….[5.4]

Now the j^{th} element of the gradient will would be

$$\left[\nabla F(x)\right]_j = \delta F(x)/\delta S_j = 2\sum_{i=1}^{n} V_i(x)\delta v_i(x)/\delta x_j$$ ……[5.5]

This gradient can be written in Matrix form as follows:
$\nabla F(x) = 2J^T(x)\, v(x)$ …….[5.6]
And J(x) is defined as the Jacobian Matrix.

In the Next step we consider the Hessian matrix. The kj element of Hessian matrix would be defined as:

$$\left[\nabla^2 F(x)\right]_{kj} = \delta^2 F(x)/\delta x_k\, \delta x_j$$ ..[5.7]

The matrix form of Hessian matrix can then be expressed
$\nabla^2 F(x) = 2\,J^T(x)\,J(x) + 2\,S(x)$…….[5.8]
Where S(x) is

$$S(x) = \sum_{i=1}^{n} V_i(x).\nabla^2 v_i(x)$$

Assuming that S(x) is small, the Hessian matrix is approximately defined as:

$\nabla^2 F(x) \equiv 2J^T(x)\,J(x)$……..[5.9]

After Substituting the values of ∇2F(x) & ∇F(x), we obtain the Gauss-Newton method as:

$X_{k+1} = X_k - [J^T(X_k)\,J(X_k)]^{-1} J^T(X_k)\,V(X_k)$….[5.10]

PSNR = 20 log 10 [255/sqrt (MSE)] ……..[5.2]

The problem with the Gauss-Newton method over the standard Newton's method is that the matrix $H=J^TJ$ may not be invertible. To overcome this problem the following modification is require to the approximate Hessian matrix:

$G = H + \mu I$.

This leads to **Levenberg-Marquardt** algorithm:

$X_{k+1} = X_k - [J^T(X_k)\,J(X_k) + \mu_{kI}]^{-1} J^T(X_k)\,V(X_k)$

Or $\Delta X_k = -[J^T(X_k)\,J(X_k) + \mu_{kI}]^{-1} J^T(X_k)\,V(X_k)$

The useful feature of this algorithm is as μk is increased it approaches the steepest descent algorithm (Gauss-Newton

Algorithm) with small learning rate. The number of iterations of the Levenberg-Marquardt back propagation algorithm described as follows:

1. Insert all inputs to the network and compute the corresponding network layers outputs and the errors through the folmula "$e_q = t_q - a^M_q$". Calculate the sum of squared errors over all inputs F(x) as:

$$F(x) = \sum e_q^T e_q = \sum\sum(e_{j,q})^2 = \sum(v_i)^2 \quad \text{.....[6.1]}$$

2. Calculate the sensitivities with the help of recurrence relation. Compute the Jacobian matrix value. Augmented the individual matrices into the Margquardt sensitivities.

3. Obtain the value of $\Delta X_{k..}$

4. Now recomputed the sum of squared errors using the xk + $\Delta X_{k..}$ If this new sum is smaller than that of computed the equation [6.1] then divide μ by v, let $X_{k+1} = X_k + \Delta X_k$ and go back to step 1. If the sum of squares is greater, then multiply μ by v and go back to step 3.

Training Procedure

During the training process data from a representative image(input image) or a class of images is encoded into a structure of the hidden layer neurons and output weight matrices. It is assumed that an image, I, used in training of size Rx C may consists of rxc blocks. The training procedure steps are following:

1. To convert a block matrix I into a matrix I' of size P x N having the training vector, x(n), formed from image blocks. As:

$$P = r.c \;\&\; p.N = R.C \text{.....[6.2]}$$

2. The target data is made equal to the output data, as:

$$D = I' \text{.....[6.3]}$$

3. The ANN is trained until the value of mean squared error, MSE, is sufficiently small.

The matrices W^i and W^j will be used for the image encoding and decoding steps. These steps are as follows:

Image Encoding

Half of the hidden layer is use for the image encoding process as:

$I \rightarrow I', H = (W^h . I')$ ……..[6.4]

Where I' is the encoded image of I.

Image Decoding

The rest half of the hidden layer of ANN is use for the image decoding procedure as:

$$Y = (W^y . H), Y \rightarrow I$$

VI. Experimental Result and Performance Assessment

Consider a Baboon image. Plots are defined for the 24 epochs.

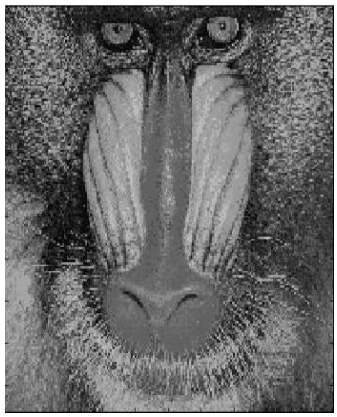

Figure 4: Original Baboon image

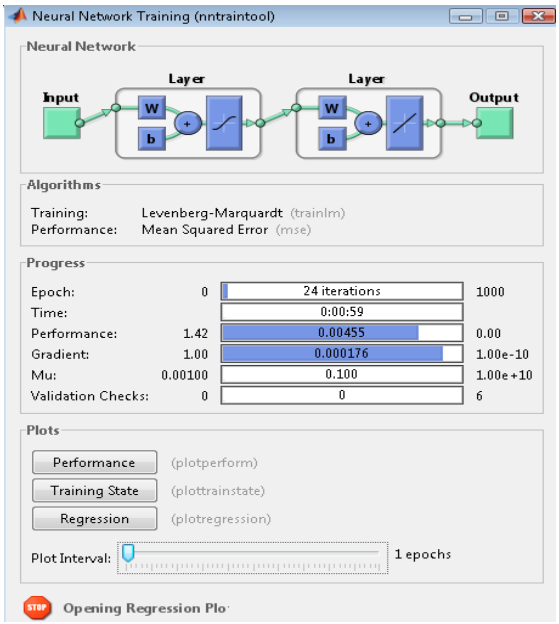

Figure 5: Neural Network Training screen

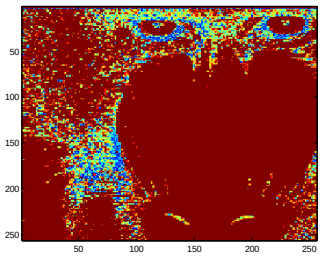

Figure 6: Decompressed image on 24 iterations

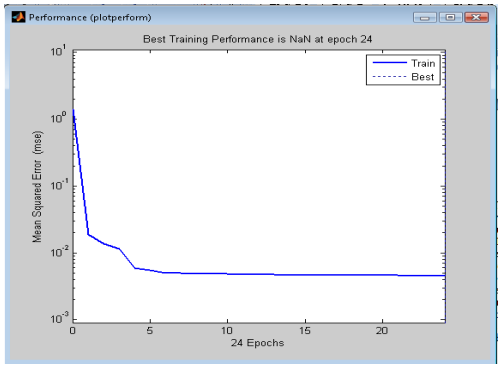

Figure 7: Performance graph for 24 iterations

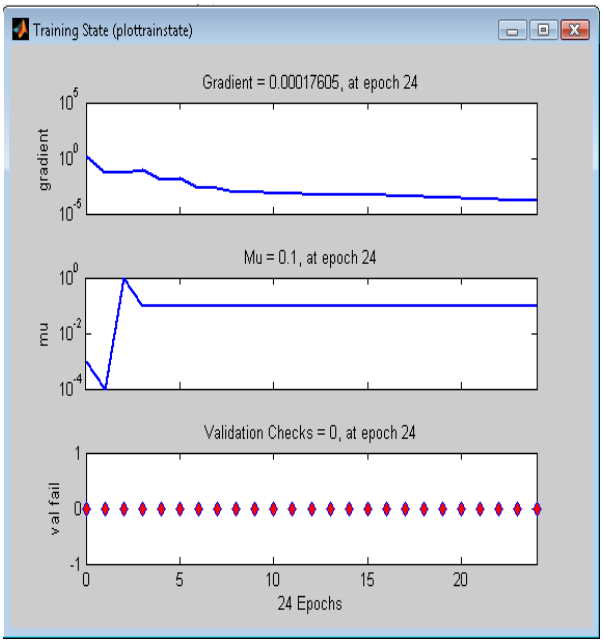

Figure 8: Training graph for 24 iterations

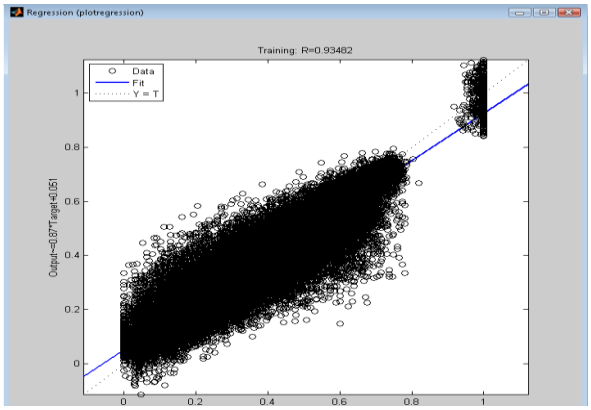

Figure 9: Regression graph

From the results, it is observed that the decompressed image is distorted due to the loss of data during computation. It is observed that the Bipolar back propagation Coding with linear scaling algorithm gives better results than statistical PCA method. But in Bipolar Coding, if the size of data set is large (Baboon image),then iterations stops without minimizing the error. This occurs due to the saturation of the network. Also, it requires the large memory area to store the iterated results. Such type of problem of neural network learning can be overcome by Levenberg-Marquardt algorithm. The results further can be improved and errors can be minimized by applying the Levenberg-Marquardt algorithm.

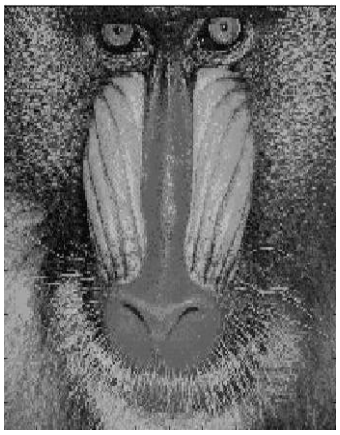

Figure 10: Outputed Baboon Image

The momentum factor and learning rate parameter decides the speed and error of the feed forward back propagation neural network. If momentum factor and learning rate is small then training of the neural network is slow and therefore required more time for compression.

Table 1: Comparison of execution time required for Bipolar Coding and Levenberg-Marquardt techniques.

Epoch	Time(Min:Sec:dig)		
	PCA	Bipolar Coding	LM
100	0:13:04	0:10:25	0:02:53
200	0:17:07	0:12:36	0:08:11
300	0:21:15	0:14:07	0:11:08
400	0:32:51	0:18:44	0:15:26
500	0:41:20	0:23:04	0:18:04
600	0:48:36	0:37:07	0:21:56
800	1:10:15	1:05:10	0:28:15
1000	1:25:53	1:21:14	0:35:10

VII. CONCLUSION

In this paper, the proposed technique is used for image compression. The algorithm is tested on varieties of benchmark images. Simulation results for standard test images with different sizes are presented. These results are compared with existing technique. Several performance measures are used to test the reconstructed image quality. It can be inferred from experimental results as shown that the proposed method performed well and results higher compression ratio. Besides higher compression ratio it also preserves the quality of the image.

It can be concluded that the integration of classical with soft computing based image compression enables a new way for achieving higher compression ratio.

VIII. ACKNOWLEDGEMENTS

The author would like to express her thanks to the management and department of ECE, Kanpur Institute of Technology for their support and encouragement during this work. She also wants to greet Mr. Pratyush Tripathi (Prof., ECE & Guide of the Project).

REFERENCES

[1] Vilas H. Gaidhane1, Vijander Singh1, Yogesh V. Hote2, Mahendra Kumar3, "*New Approaches for Image Compression Using Neural Network*" , Journal of Intelligent Learning Systems and Applications," Published Online November 2011, 3, 220-229.

[2] B.Arunapriya , D.KavithaDevi, "*Improved Digital Image Compression using Modified Single Layer Linear Neural Networks" ,* International Journal of Computer Applications (0975 – 8887) Volume 10– No.1, November 2010.

[3] K.V.V.Kumar1*, J.Supriyanka2, K.Rajkamal1, Venkata Raviteja .K1, G.Manoj Kumar1, J.V.Suresh1, M.Rakesh Babu3,"*Fuzzy Neural Network based Image Compression using Levenberg-Marquardt Algorithm*",International Journal of Science and Advanced Technology (ISSN 2221-8386), A.P.,India,Volume 2 ,January 2012.

[4] V. Gaidhane, V. Singh and M. Kumar, "*Image Compression Using PCA and Improved Technique with MLP Neural Network*," Proceedings of IEEE International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, 16-17 OctoBer 2010,pp. 106-110.

[5] S. N. Sivanandam, S. Sumathi and S. N. Deepa, "*Introduction to Neural Network Using MATLAB 6.0*," 2nd dition, Tata Mc-Graw Hill Publication, Boston, 2008.

[6] http://en.wikipedia.org/wiki/Neuralnetwork.

[7] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Reading. MA: Addison Wesley, 2004.

[8] A. Rahman, Chowdhury Mofizur Rahman, "*A New Approach for Compressing Color Images using Neural Network*", Proceedings of International Conference on Computational Intelligence for Modeling,Control and Automation – CIMCA 2003 ,Vienna, Austria, 2003.

[9] N.Sonehara, M.Kawato, S.Miyake, K.Nakane, *"Image compression using a neural networkmodel"*,International Joint Conference on Neural Networks, Washington DC,1989.

[10] D. Taubman: *High performance scalable image compression with EBCOT*, IEEE Transactions on Image Processing, July 2000, 1158–1170.

AUTHORS

First Author – Prachi Tripathi, Master in Technology(ECE), Kanpur Institue of Technology, dhoolika77@gmail.com