Artificial Neural Network based Image Compression using Levenberg-Marquardt Algorithm

Pranob K Charles[1],Dr. H.Khan[2],Ch.Rajesh Kumar[3],N.Nikhita[3]
Santhosh Roy[3],V.Harish[3],M.Swathi[3]
[1]*Associate professor, Department of ECE, K.L.University, Guntur, A.P, India*
[2]*Professor*,HOD, *Department of ECE, K.L.University, Guntur, A.P, India*
[3]*Project Students , Department of ECE, K L University, Guntur, A.P, India*

ABSTRACT
Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies._Image compression is one of the popular image processing technologies. We have used an adaptive method for image compression based on complexity level of the image and modification on levenberg-marquardt algorithm for MLP neural network learning is used. In this method different back propagation artificial neural networks are used as compressor and de-compressor and it is achieved by dividing the image in to blocks, computing the complexity of each block and then selecting one network for each block according to its complexity value. The algorithm used has good convergence. This reduces the amount of oscillation in learning procedure. To realise this method practically, multilayer neural (input,hidden and output layers) networks are used.

Keywords – **Artificial neural network, MLP, Training, Compression, Complexity.**

I. INTRODUCTION
Neural networks are inherent adaptive systems, they are suitable for handling non stationeries in image data. Artificial neural network can be employed with success to image compression. The greatest potential of neural networks is the high speed processing that is provided through massively parallel VLSI implementations. The choice to build a neural network in digital hardware comes from several advantages that are typical for digital systems. The crucial problems of neural network hardware are fast multiplication, building a large number of connections between neurons, and fast memory access of weight storage or nonlinear function look up tables.

The most important part of a neuron is the multiplier, which performs high speed pipelined multiplication of synaptic signals with weights. As the neuron has only one multiplier the degree of parallelism is node parallelism. Each neuron has a local weight ROM (as it performs the feed-forward phase of the back propagation algorithm) that stores, as many values as there are connections to the previous layer.

An accumulator is used to add signals from the pipeline with the neuron's bias value, which is stored in an own register.

The aim is to design and implement image compression using Neural network to achieve better SNR and compression levels. The compression is first obtained by modeling the Neural Network in MATLAB. This is for obtaining offline training.

II. NEURAL NETWORKS
2.1 Artificial Neural Network
An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well. Artificial Neural Network is a system loosely modeled on the human brain. The field goes by many names, such as connectionism; parallel distributed

processing, neurocomputing, natural intelligent systems, machine learning algorithms, and aritificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections.

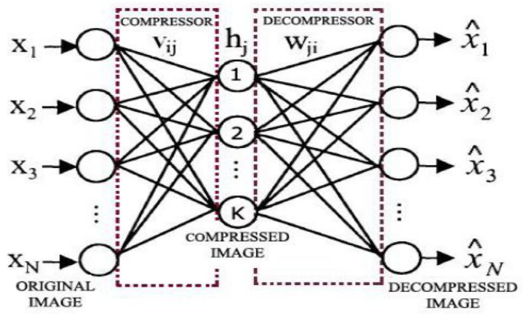


Fig.1.1Basic compression structure

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. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

This expert can then used to provide projections given new situations of interest and answer "what if" questions.

The advantages include Adaptive learning, Self-Organization,Real Time Operation,Fault Tolerance via Redundant Information Coding.

2.2 Layers

Artificial neural network are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers, which are then connected to one another. How these layers connect may also vary. Basically, all artificially neural networks have a similar structure of topology. Some of the neurons interface the real world to receive its inputs and other neurons provide the real world with the network's outputs. All the rest of the neurons are hidden form view. The input layer consists of neurons that receive input form the external environment. The output layer consists of neurons that communicate the output of the system to the user or external environment. There are usually a number of hidden layers between these two layers; the fig2 below shows a simple structure with only one hidden layer.

When the input layer receives the input its neurons produce output, which becomes input to the other layers of the system.

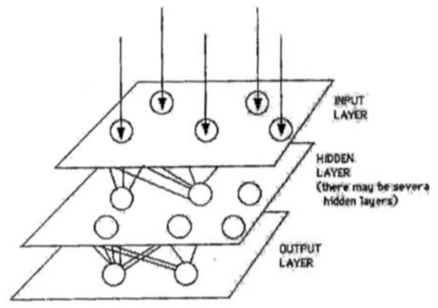


Fig.2.1 *THREE peceptron for image compression*

The process continues until a certain condition is satisfied or until layer is invoked and fires their output to the external environment. To determine the number of hidden neurons the network should have to perform its best, one are often left out to the method trial and error. If the hidden number of neurons are increased too much an over fit occurs, that is the net will have problem to generalize. The training set of data will be memorized, making the network useless on new data sets.

2.3 Learning

The brain basically learns from experience. Neural networks are sometimes called machine-learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge but adjusting these connection weights. The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training. The training method usually consists of one of three schemes:

2.3.1. Unsupervised learning.

Used no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning. From Human Neurons to Artificial Neuron other aspect of learning concerns the distinction or not of a separate phase, during which the network is trained, and a subsequent operation phase. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line. The hidden neurons must find a way to organize themselves without help from the outside. In this approach, no sample outputs are provided to the network against which it can measure its predictive performance for a given vector of inputs. This is learning by doing.

2.3.2. Reinforcement learning

This incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms is the least mean square (LMS) convergence**.** This method works on reinforcement from the outside. The connections among the neurons in the hidden layer are randomly arranged, then reshuffled as the network is told how close it is to solving the problem. Reinforcement learning is also called supervised learning, because it requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results.

Both unsupervised and reinforcement suffers from relative slowness and inefficiency relying on a random shuffling to find the proper connection weights.

2.3.3. Back propagation

This method is proven highly successful in training of multilayered neural nets. The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance. A form of supervised learning.

2.4 Image Compression

Direct transmission of the video data requires a high-bit-rate (Bandwidth) channel. When such a high bandwidth channel is unavailable or not economical, compression techniques have to be used to reduce the bit rate and ideally maintain the same visual quality. Similar arguments can be applied to storage media in which the concern is memory space. Video sequence contain significant amount of redundancy within and between frames. It is this redundancy frame. It is this redundancy that allows video sequences to be compressed. Within each individual frame, the values of neighboring pixels are usually close to one another. This spatial redundancy can be removed fro the image without degrading the picture quality using "Intraframe" techniques.

Also, most of the information in a given frame may be present in adjacent frames. This temporal redundancy can also be removed, in addition to the "within frame" redundancy by "interframe" coding.

III. PROPOSED IMAGE COMPRESSION USING NEURAL NETWORKS (LM ALGORITHM)

3.1 Introduction

A two layer feed-forward neural network and the Levenberg Marquardt algorithm was considered. Image coding using a feed forward neural network consists of the following steps:

An image, F, is divided into rxc blocks of pixels. Each block is then scanned to form a input vector x (n) of size

$$p = r \times c \qquad \ldots\ldots\ldots\ldots\ldots\ldots [3.1]$$

It is assumed that the hidden layer of the layer network consists of L neurons each with P synapses, and it is characterized by an appropriately selected weight matrix $\mathbf{W_h}$. All N blocks of the original image is passed through the hidden layer to obtain the hidden signals, h(n), which represent encoded input image blocks, x(n) If L<P such coding delivers image compression.

It is assumed that the output layer consists of m=p=rxc neurons, each with L synapses. Let W_y be an appropriately selected output weight matrix. All N hidden vector h(n), representing an encoded image H, are passed through the output layer to obtain the output signal, y(n). The output signals are reassembled into p=rxc image blocks to obtain a reconstructed image, F_r. There are two error matrices that are used to compare the various image compression techniques. They are Mean Square Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image whereas PSNR is the measure of the peak error.

$$MSE = {}^{1}\!/_{MN} \sum_{y=1}^{m} \sum_{x=1}^{n} [I(x,y) - I'(x,y)]^2$$

…………

[3.2]

The quality of image coding is typically assessed by the Peak signal-to-noise ratio (PSNR) defined as

PSNR = 20 log $_{10}$ [255/sqrt(MSE)] …………

[3.3]

Training is conducted for a representative class of images using the Levenberg Marquardt algorithm. Once the weight matrices have been appropriately selected, any image can be quickly encoded using the W_h matrix, and then decoded (reconstructed) using the W_y matrix.

3.2Basic Algorithm:

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

$X_{k+1} = X_k - A_k^{-1} g_k$, …………..

[3.4]

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

It is assume d that F (x) is a sum of squares function:

$$F(x) = \sum_{r=1}^{n} v_i^2(x) = V^T(x) v(x)$$

………….[3.5]

Then the jth element of the gradient will would be

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

…………[3.6]

The gradient can be written in matrix form:

$\nabla F(x) = 2J^T(x)\, v(x)$
………………[3.7]

Where J(x) is the Jacobian matrix.

Next the Hessian matrix is considered. The k.j element of Hessian matrix would be

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

…..…[3.8]

The Hessian matrix can then be expressed in matrix form:

$\nabla^2 F(x) = 2 J^T(x) J(x) + 2 S(x)$; where

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

Assuming that S(x) is small, the Hessian matrix is approximated as

$\nabla^2 F(x) \equiv 2 J^T(x) J(x)$
……………[3.9]

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

$X_{k+1} = X_k - [J^T(X_k) J(X_k)]^{-1} J^T(X_k) V(X_k)$
...….[3.10]

One problem with the Gauss-Newton over the standard Newton's method is that the matrix $H=J^TJ$ may not be invertible. This can be overcome by using the following modification to the approximate Hessian matrix:

G = H + μ I.

This leads to Levenberg –Marquardt algorithm

$X_{k+1} = X_k - [J^T(X_k) J(X_k) + \mu_{kI}]^{-1} J^T(X_k) V(X_k)$…….[3.11] this algorithm has the very useful feature that as μk is increased it approaches the steepest descent algorithm with small learning rate.

3.3Training Procedure

During training procedure data from a representative image or a class of images is encoded into a structure of the hidden and output weight matrices. It is assumed that an image, F, used in training of size Rx C and consists of rxc blocks.

1. The first step is to convert a block matrix F into a matrix X of size P x N containing training vectors, x(n), formed from image blocks.That is:

$$P= r.c \text{ and } p.N = R.C$$

2. The target data is made equal to the data, that is: D=X

3. The network is then trained until the mean squared error, MSE, is sufficiently small. The matrices W^h and W^y will be subsequently used in the image encoding and decoding steps.

Image Encoding: The hidden-half of the two-layer network is used to encode images. The Encoding procedure can be described as follows:

$$F \rightarrow X, \quad H= (W^h. X) \quad \text{…………………[3.12]}$$

Where X is an encoded image of F.

Image Decoding: The image is decoded (reconstructed) using the output-half the two-layer network. The decoding procedure is described as follows:

$$Y = (W^y. H), \quad Y \rightarrow F \quad \text{…………………[3.13]}$$

3.4 Algorithm

Step1: Read the test image

Step2: Divide the image into blocks of pixels.

Step3: Scan each block for the complexity level.

Step4: Initialize the neurons.

Step5: Apply scanned vectors to each neuron on the input layer.

Step6: Depending on the weights and the logic involved, perform the operations (TRANSIG).

Step7: Pass them to the hidden layer.

Step8: Again, the same as in step6 (PURELIN).

Step9: Reassemble the outputs.

Step10: Train the neural network and remain the weights.

IV. MATLAB RESULTS AND GRAPHS

In this section, simulation results for different images (64x64) are shown. Their performance measure graphs are also included. Considered the images cameraman.tif and fabric.png form the MATLAB library.

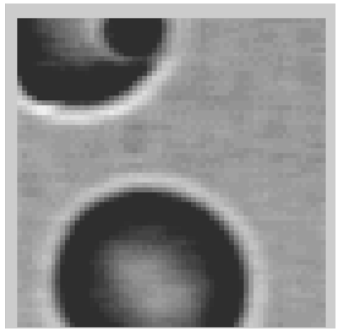


Fig3.1 Original image-1

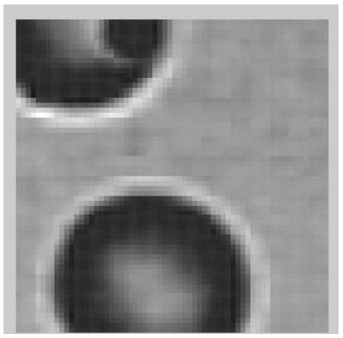


Fig3.2 Encoded (compressed) image-1

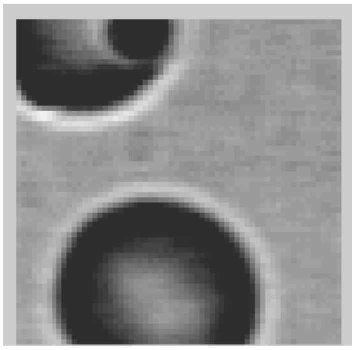


Fig3.3 Decoded (Decompressed) image-2

The performance measure of the applied method can be observed from the graph fig3.4 shown.

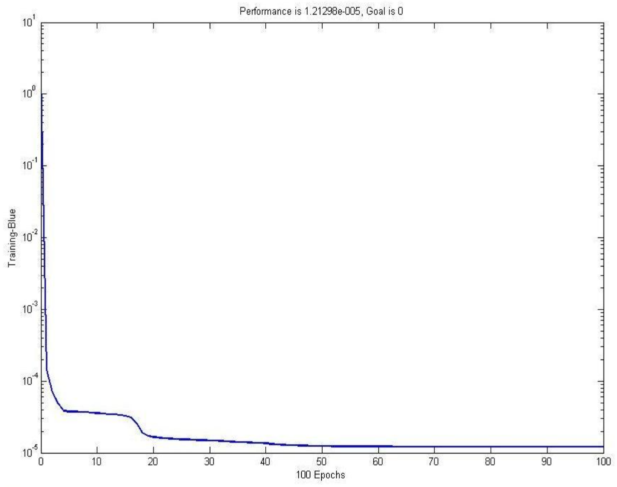


Fig3.4 performance plot

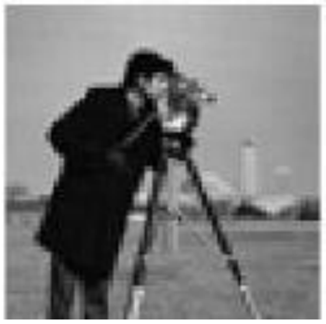


Fig3.5original image-2

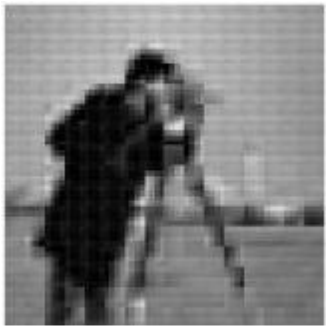


Fig3.6 Compressed image-2

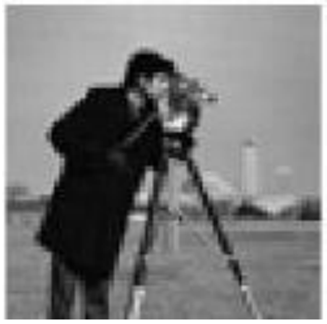


Fig 3.7 Decompressed image-2

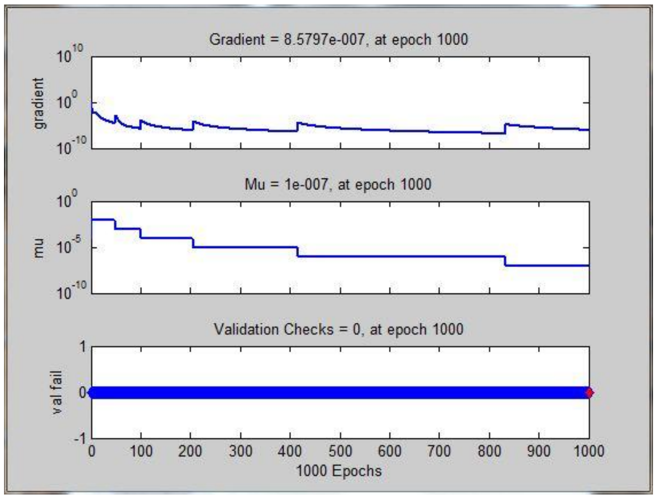


Fig3.8 training state graph

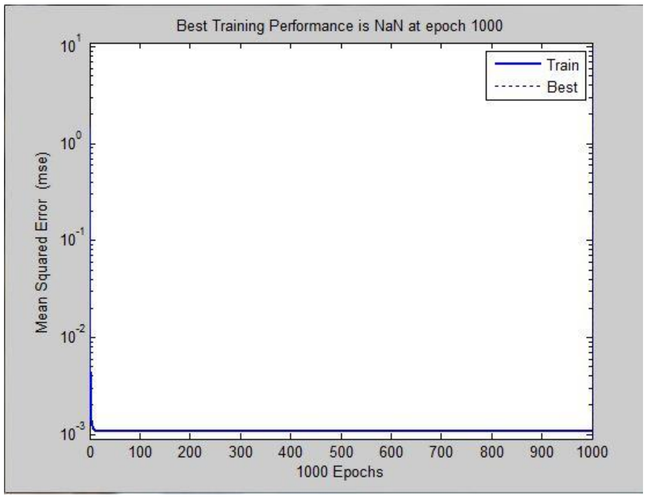


Fig3.9 Performance plot

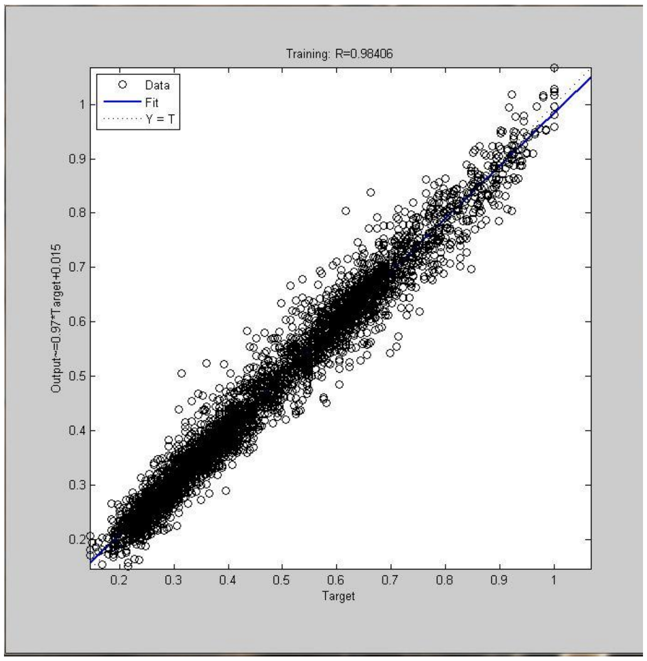
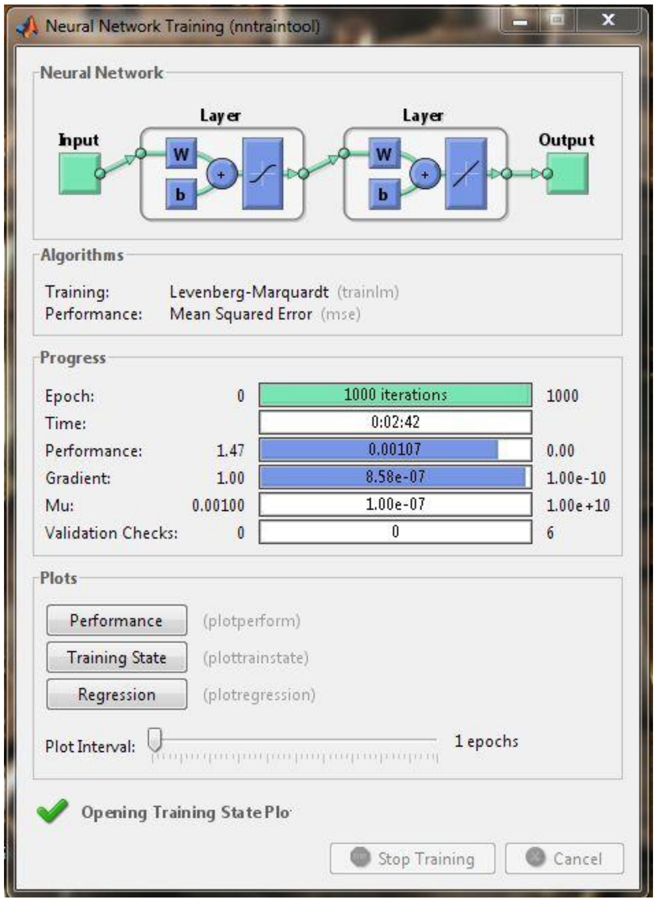


Fig3.10 Regression plot

The default training tool for neural networks provided in MATLAB is fig3.11 below.



Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	1000 iterations	1000
Time:		0:02:42	
Performance:	1.47	0.00107	0.00
Gradient:	1.00	8.58e-07	1.00e-10
Mu:	0.00100	1.00e-07	1.00e+10
Validation Checks:	0	0	6

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs

Opening Training State Plo

Fig3.11 Neural network training tool(MATLAB)

V. Conclusion

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture. Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain. Perhaps the most exciting aspect of neural networks is the possibility that some day 'conscious' networks might be produced. There is a number of scientists arguing that consciousness is a 'mechanical' property and that 'conscious' neural networks are a realistic possibility.

Even though neural networks have a huge potential we will only get the best of them when they are integrated with computing, AI, fuzzy logic and related subjects. Neural networks are performing successfully where other methods do not, recognizing and matching complicated, vague, or incomplete patterns.

ACKNOWLEDGEMENTS

The authors like to express their thanks to the management and department of ECE, K L University for their support and encouragement during this work.

REFERENCES

[1] P.J. BURT, E.H. ADELSON, THE LAPLACIAN PYRAMID AS A COMPACT IMAGE CODE, IEEE TRANS. ON COMMUNICATIONS, PP. 532–540, APRIL 1983.

[2] D. SHAO, L.A. MATEOS, W.G. KROPATSCH, IRREGULAR LAPLACIAN GRAPH PYRAMID,CVWW 2010.

[3] S. BECKER AND M. PLUMBLEY, "UNSUPERVISED NEURAL NETWORK LEARNING PROCEDURES FOR FEATUREEXTRACTION AND CLASSIFICATION," 1996, TO APPEAR *INT. J. APPLIED INTELLIGENCE*.

[4] G. W. COTTRELL AND P. MUNRO, "PRINCIPAL COMPONENTS ANALYSIS OF IMAGES VIA BACK PROPAGATION," IN *SPIE VOL. 1001 VISUAL COMMUNICATIONS AND IMAGE PROCESSING '88*, 1988, PP. 1070–1077.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York, NY: Macmillan, 1994.

[6] A. N. Netravali and J. O. Limb, "Picture coding:A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366–406, March 1980.

[7] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, no. 3, pp. 349– 389, March 1981.

[8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed. San Diego, CA: Academic Press, 1982, vol. I & II.

[9] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice- Hall, 1984.

[10] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523–548, April 1985.

[11]A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York,NY: Plenum Press, 1988.

[12]A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic Publishers, 1992.

[13]J. A. Storer and M. Cohn, Eds., *Proc. Data Compression Conference*. Snowbird, UT: IEEE Computer Society Press, March 30 - April 2,1993.

[14]N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, no. 10, pp. 1385–1421, October 1993.

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

[16]R. C. Gonzales, R. E. Woods, "Digital Image Processing", Second Edition, Prentice-Hall, 2002.

[17] Veisi H., Jamzad M., "Image Compression Using Neural Networks", Image Processing and Machine Vision Conference (MVIP), Tehran, Iran, 2005.

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

[19]G.L. Sicuranza, G. Ramponi, S. Marsi, "Artificial neural network for image compression", Electronic letters 26, 477- 479, 1990.

[20]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