ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

VLSI Architecture and implementation for 3D Neural Network based image compression

Deepa.S[1], Dr.P.Cyril Prasanna Raj [2], Dr.M.Z.Kurian[3], Manjula.Y[4]

1-Final year M.Tech VLSI & embedded systems deepa2292959@gmail.com,

2-Professor, member of IEEE cyrilyahoo@gmail.com,

3- Dean,HOD,E&C,SSIT,Tumkur mzkurian@gmail.com,

4-Lecturer,Dept of E&C,SSIT,Tumkur manjulayarava@gmail.com

Abstract: **Image compression is one of the key image processing techniques in signal processing and communication systems. Compression of images leads to reduction of storage space and reduces transmission bandwidth and hence also the cost. Advances in VLSI technology are rapidly changing the technological needs of common man. One of the major technological domains that are directly related to mankind is image compression. Neural networks can be used for image compression. Neural network architectures have proven to be more reliable, robust, and programmable and offer better performance when compared with classical techniques.**
In this work the main focus is on development of new architectures for hardware implementation of 3-D neural network based image compression optimizing area, power and speed as specific to ASIC implementation, and comparison with FPGA.

Key words: Image compression, 3-D neural network, FPGA, ASIC

I. INTRODUCTION

Neural network for image compression and decompression have been adopted as they achieve better compression and also work in noisy environment. Many approaches have been reported in realizing the Neural Network architectures on software and hardware for real-time applications. Today's technological growth, has led to scaling of transistors and hence complex and massively parallel architecture are possible to realize on dedicated hardware consuming low power and less area. This work reviews the neural network approaches for image compression and proposes hardware implementation schemes for Neural Network. The transport of images across communication paths is an expensive process. Image compression provides an option for reducing the number of bits in transmission. This in turn helps increase the volume of data transferred in a space of time, along with reducing the cost required. It has become increasingly important to most computer networks, as the volume of data traffic has begun to exceed their capacity for transmission. Traditional techniques that have already been identified for data compression include: Predictive Coding, Transform coding and Vector Quantization. In brief, predictive coding refers to the decorrelation of similar neighboring pixels within an image to remove redundancy. Following the removal of redundant data, a more compressed image or signal may be transmitted. Transform-based compression techniques have also been commonly employed. These techniques execute transformations on images to produce a set of coefficients. A subset of coefficients is chosen that allows good data representation (minimum distortion) while maintaining an adequate amount of compression for transmission. The results achieved with a transform based technique is highly dependent on the choice of transformation used (cosine, wavelet, etc.). Finally vector quantization techniques require the development of an appropriate codebook to compress data. Usages of codebooks do not guarantee convergence and hence do not necessarily deliver infallible decoding accuracy. Also the process may be very slow for large codebooks as the process requires extensive searches through the entire codebook.

Artificial Neural Networks (ANNs) have been applied to many problems and have demonstrated their superiority over traditional methods when dealing with noisy or incomplete data. One such application is for image compression. Neural Networks seem to be well suited to this particular function, as they have the ability to preprocess input patterns to produce simpler patterns with fewer components. This compressed information (stored in

a hidden layer) preserves the full information obtained from the external environment. Not only can ANN based techniques provide sufficient compression rates of the data in question, but security is easily maintained. This occurs because the compressed data that is sent along a communication line is encoded and does not resemble its original form.

II. SYSTEM DESIGN

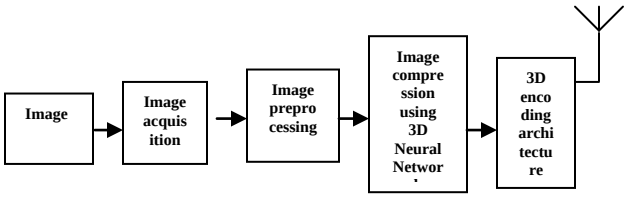


Fig 1: Block diagram overview

The fig 1 gives the overall view of the work. Here image may be any picture or a photograph or a visual data. This is captured or acquired by an optical lence such as a camera. The image is "discretized" i.e., defined on a discrete grid and stored as two-dimensional collection of light values (or gray values).The image preprocessing does the function of adjustment of pixels and removal of noise from the captured image. The image is compressed using 3D Neural Network architecture and encoded for image transmission.

The 3D Neural Network architecture for image compression is main topic of interest of this project work. An attempt to design and develop 3-D adaptive neural network architecture for image compression and decompression that can work in real time, noisy environment is made here.

The proposed 3-DNN architecture is as shown in figure 2. Typical way of compression with neural network is using hidden layer with lower dimension then input or output layer. In this case, network input and output is an image. The input and hidden layers perform the compression, it transforms the input vector of dimension *d(IN)* to hidden layer space of dimension *d(HI).* The dimension of hidden layer is lower than of input layer *d(HI)<d(IN)*. Output layer with the help of hidden layers performs decompression. It transforms the vector from hidden layer with lower dimension to output vector space with higher dimension *d(HI)<d(OUT).* The dimension of input and output layer is the same *d(IN)=d(OUT).*

The basic architecture for image compression using neural network is shown in figure 3

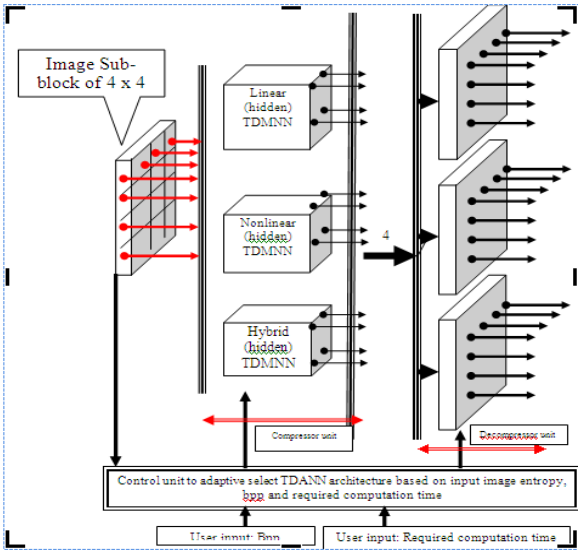


Fig 2: Proposed 3DNN architecture

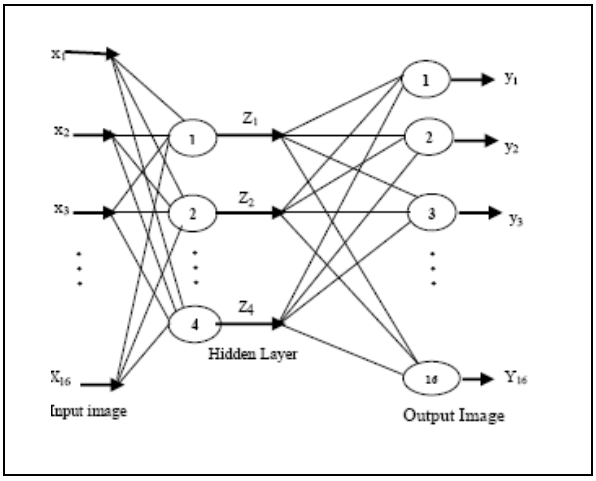


Fig 3: Basic architecture for image compression using NN

M-dimensional coefficient vector y is calculated as

$Y = Z \times X$

then the inverse transformation is given by the transpose of the forward transformation matrix resulting in the reconstructed vector

$X = Z^T \times Y$

Here 16 input neurons, 4 neurons each in 4 hidden layers and 16 output neurons are considered. purlin function and purlin functions are used to train the input and output neurons of hidden layers. Back propagation technique is also used.

III. SYSTEM IMPLEMENTATION/ SIMULATION

Matlab program is written in the following steps:
The input image is read by the instruction imread. This will read image saved in the specified path. Next a part of image is read by specifying particular row

and column values. This part of the image is displayed. Rearrangement of image is the selected application of the multilayer perceptron. Next is the training procedure where specified weight and bias is obtained for image compression & decompression. The reshaping of a block matrix in to a matrix of vector occurs. The design consists of matrix multiplication of two matrices, one is the input image samples(each column), and the second is the weight matrix obtained after training. This multiplied output is added with bias to obtain the compressed output that gets transmitted or stored in compressed format. On the decompression side, the compressed data in matrix form is multiplied with the output weight matrix and added with output bias to get back the original image. The image quality of the decompressed image depends on the weight matrix and the compression ratios. The image data of size 16x16 is taken in each column wise i.e column 1 of size 16x1 then multiplied by the weight matrix of 4x16 and added with bias matrix of size 4x1 to get a compressed output of 4x1. On the decompression side 4x1 input matrix (compressed image) is multiplied with the weight matrix of size 16x4 and added with output bias matrix of size16x1to get output decompressed matrix of size 16x1. This procedure is repeated for 15 more times to reproduces the original image. In order to achieve better compression nonlinear functions are used both at transmitter and receiver section. The Neural Network has 4 hidden and 16 output layers. Purlin and purlin functions are used for input and output hidden layers. Back propagation technique is used. Finally rearrangement of the vectors in to block matrix to display the image.

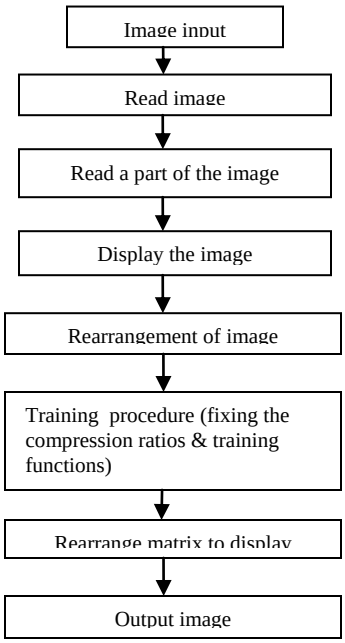


Fig 4: Flow chart of matlab program

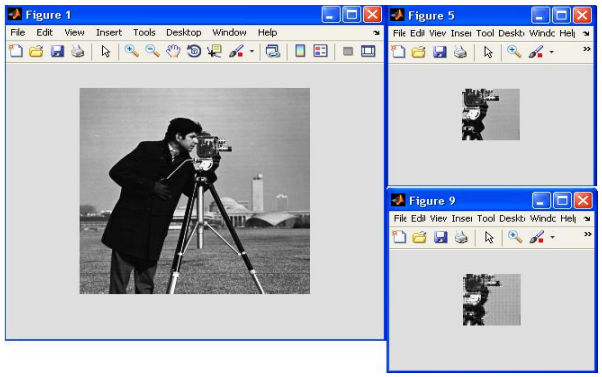
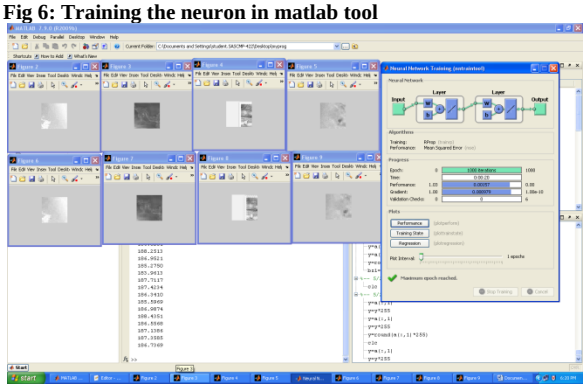


Fig 5

Here the image is read in figure. Next a part of image(64*64) is read for compression and the decompressed output is shown.

Fig 6: Training the neuron in matlab tool



The above snapshot shows training the neuron in matlab tool.

Simulation Results in matlab

The different compression algorithms can be compared based on certain performance measures. Compression Ratio (CR) is the ratio of the number of bits required to represent the data before compression to the number of bits required after compression. Bit rate is the average number of bits per sample or pixel (bpp), in the case of image. The image quality can be evaluated objectively and subjectively. A standard objective measure of image quality is reconstruction error given by equation 1.

Error E = Original Image – Reconstructed image (1)

Two of the error metrics used to compare the various image compression techniques are the mean square error (MSE) and the Peak Signal to Noise Ratio (PSNR). MSE refers to the average value of the square of the error between the original signal and the reconstruction as given by equation 2. The important parameter that indicates the quality of the reconstruction is the peak signal-to-noise ratio

(PSNR). PSNR is defined as the ratio of square of the peak value of the signal to the mean square error, expressed in decibels.

MSE = E / (SIZE OF IMAGE) (2)

The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a Σmeasure of the peak error. The mathematical formulae for the computation of MSE & PSNR is :

$$MSE = 1/MN \left[\sum_{i=1}^{M} \sum_{j=1}^{N} (Ixy - I'xy)^2 \right] \quad (3)$$

$$PSNR = 20 * \log (255 / \text{sqrt}(MSE)) \quad (4)$$

where I(x,y) is the original image, I'(x,y) is the approximated version (which is actually the decompressed image) and M, N are the dimensions of the images, 255 is the peak signal value. A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR. Higher values of PSNR produce better image compression because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, a compression scheme having a lower MSE (and a high PSNR), can be recognized as a better one.

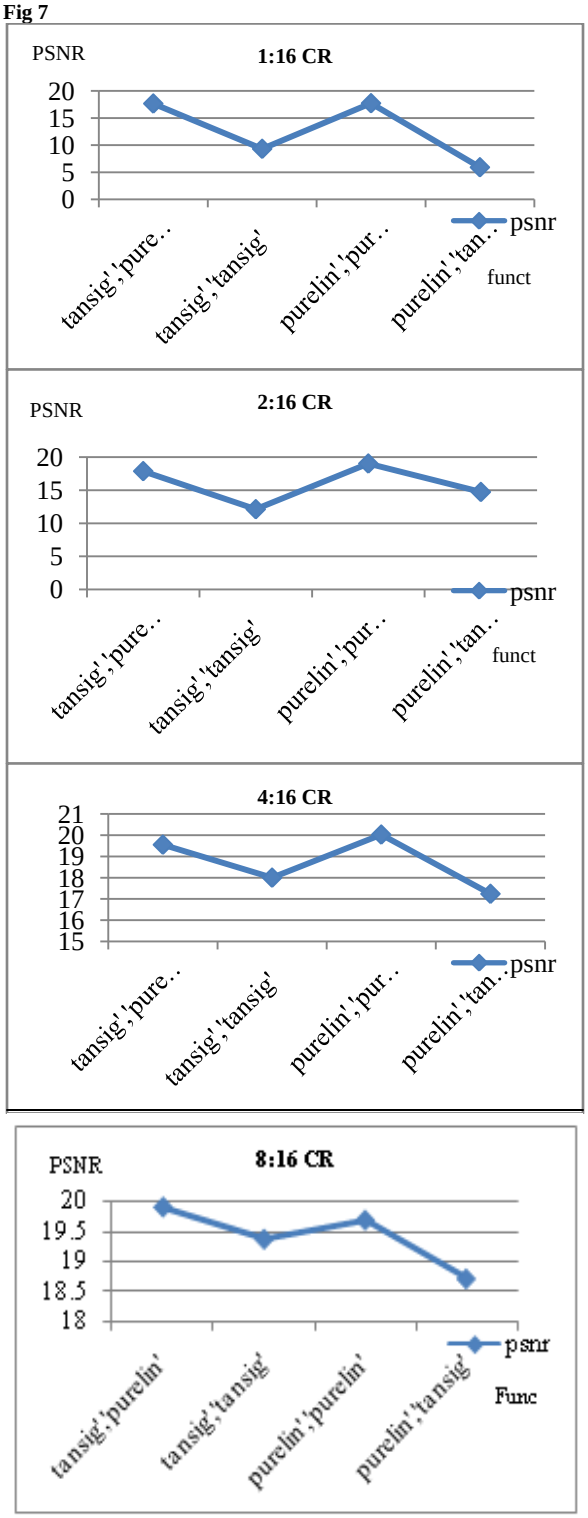
Compression results by adaptive BPNN Structure for size=64*64,r=4,CR=4:16(75%)

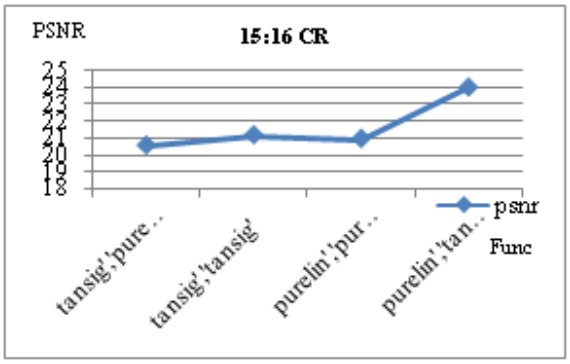
Sl no.	Test Image	Image max Error	Image MSE	PSNR
1	cameraman	108	262.01	23.94
2	board	166	958.40	17.7765
3	cell	38	26.462	33.9045
4	circuit	15	14.113	36.6344
5	Lena	61	100.00	28.1305
6	sun	44	115.84	27.2919
7	girl	58	42.806	31.8157
8	Blue hills	8	22.182	44.7420
9	Sunset	55	39.371	32.1790
10	Water lilies	72	31.379	33.164
11	Winter	36	47.626	31.352
12	Drawing	255	170.77	15.194
13	College	57	97.780	24.037
14	Garden	78	87.990	25.000
15	My photo	101	197.22	22.991
16	Holi	97	175.34	25.7
17	Bhagavadgeetha	67	65.63	29.96
18	Devinecouple	74	80.20	29.1
19	krishna	39	7.1	39.6
20	Goddess	65	45.048	31.5940

Table 1: COMPRESSION RESULTS

Combination of different transfer functions gives different decompressed images for the same input image. Combination of different compression ratios (CR)with different transfer functions gives different PSNR values for the same input image.

Fig 7



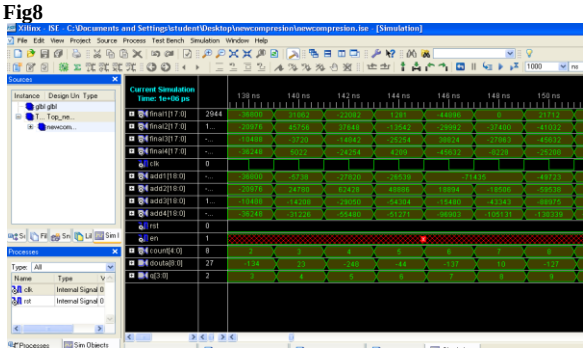


For a better image, the max error must be as low as possible, the mean square error(MSE) must also be small and the peak signal to noise ratio must be high. From the above observations the 'tansig','purelin' and 'purelin','purelin' functions give consistently high PSNRs' for the respective compression ratios(CR). Trainrp is a network training function used in this work that updates weight and bias values according to the resilient backpropagation algorithm (Rprop).Trainlm may also be used which is also a network training function that updates weight and bias values according to Levenberg-Marquardt optimization but consumes huge amount of memory. In this work back propagation technique is used. The above results are for 64*64 image. The NN with BP algorithm is also trained with 128*128, 256*256 and 521*512 images. The network trained using backpropagation algorithm is realized using multipliers and adders for FPGA realization .The work is being carried out with the design of fast multiplier which uses Wallace technique. A 9 bit Wallace multiplier and a 18 bits carry save adder is designed for this project work.

The inputs for 9 bit Wallace tree multiplier and 18 bits carry save adder is obtained by the MATLab program.HDL coding is being carried out for FPGA realization. A 3D Neural network algorithm is planned for the design.

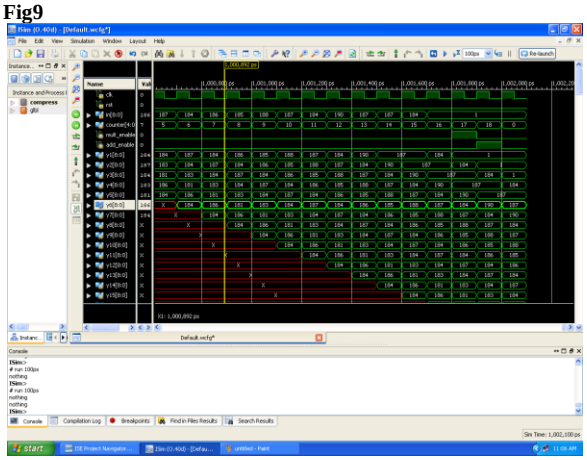
HDL/VHDL and FPGA simulation results

Fig8



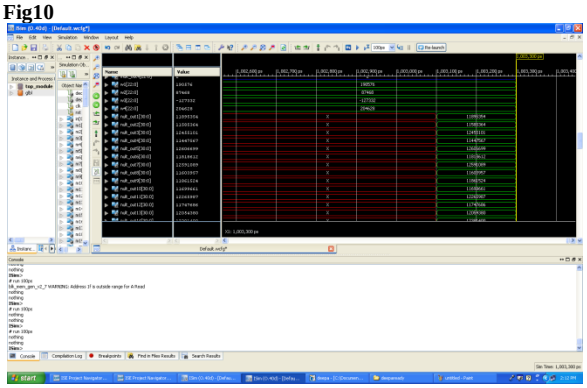
output of 9 bits Wallace multiplier and 18 bits carry save adder

Fig9



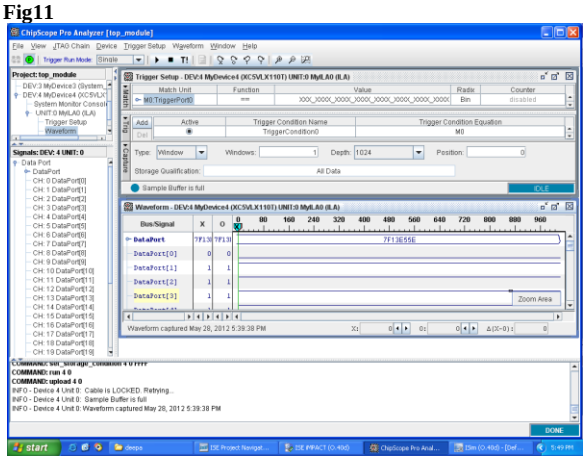
Inputs(inputs with scaling) are stoered as serial in parallel out registers

Fig10



output of compression and decompression in HDL coding.

Fig11



Output obtained by the vertex5 chip scope design

IV.HARDWARE AND SOFTWARE REQUIREMENTS

The Neural Network archicture will be synthesized on FPGA to estimate the hardware complexity for efficient ASIC implementation. The design is mapped on VERTEX5 device from Xilinx 13.1

vertion. As the design is mapped on FPGA, it supports Reconfigurability. Reconfigurability can be achieved by changing the input layer values for better compression and decompression.

The HDL code for FPGA implementation is modified for ASIC implementation. The general coding styles is adopted for building optimized RTL code for ASIC implementation.
To support higher order multiplication it is modeled using HDL, and there is a need to develop an RTL model using efficient coding styles for ASIC Synthesis using Synopsys Design Compiler Version 2007.03. The timing analysis will be carried out using Synopsys Prime Time.

In brief the hardware used are the FPGA platform, and ASIC libraries.
The software requirements are Xilinx for HDL coding , Design compiler and Matlab.

V. ADVANTAGES AND LIMITATIONS

Advantages:
Among the all present method, neural network is of special interest due to the success it has in many applications.
1. Learning ability,
2. System identification,
3. Robustness against noise,
4. Capability of optimum estimation and being used in parallel structures
5.Various kinds of neural networks like multilayer perceptron (MLP), Hopfield, learning vector quantization (LVQ), self-organizing map (SOM) and principal component neural networks have been used to compress images. The VLSI Architecture for 3D Neural Network based image compression consume less power and space, hence suitable for low cost and reliable Hardware implementation, reconfigurable on FPGA and good time to market. The process of image compression is very fast as it uses neural network rather than codebook.

Limitations:
This technique may have small amount of distortion in the decompressed image

VI. CONCLUSION

The neural network architecture for image compressions has been analyzed on FPGA and ASIC platforms and need to implement on 3D neural network. A Matlab program is written to train the neurons for compression and decompression of image. Further this is coded in HDL for FPGA realization and the HDL code is modified for ASIC Designing and implementation. Low power techniques are used to reduce power dissipation of the complex architecture.

VII. REFERENCE

[1]. K.VenkataRamanaiah and Cyril Prasanna Raj VLSI Architecuture for Neural Network Based Image Compression, © 2010 IEEE

[2]. Hamdy Soliman Computer Science Department New Mexico Tech Neural Net Simulation: SFSN Model For Image Compression, Proceedings of the IEEE, 2001

[3]. Daniel Matolin, J¨org Schreiter, Stefan Getzlaff and Ren´e Sch¨uffny,"An Analog VLSI Pulsed Neural Network Implementation for Image", Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'04) IEEE

[4]. Arbib, Michael A. (Ed.) (1995). The Handbook of Brain Theory and Neural Networks.

[5]. Alspector, U.S. Patent 4,874,963 "Neuromorphic learning networks". October 17, 1989.

[6]. Ivan Vilovic," An Experience in Image Compression UsingNeural Networks",48th International Symposium ELMAR-2006, 07-09 June 2006, Zadar, Croatia

[7]. Hadi Veisi, Mansour Jamzad," A Complexity-Based Approach in Image Compression using Neural Networks", International Journal of Signal Processing 5-2-2009

[8]. Rafid Ahmed Khalil," Hardware Implementation of Backpropagation Neural Networks on Field programmable Gate Array (FPGA)", Al-Rafidain Engineering Vol.16 No.3 Aug. 2008

[9]. Rehna. V. J, Jeya Kumar. M. K,"Hybrid Approaches to Image Coding: A Review" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 7, 2011

[10]. Jan Pohl, Petr Polách, Václav Jirsík,"ALTERNATIVE WAY OF IMAGE COMPRESSION WITH NEURAL NETWORK",

[11]. Hossein SAHOOLIZADEH and Amir Abolfazl SURATGAR Adaptive Image Compression Using Neural Networks 5th International Conference: Sciences Of Electronics, Technologies of Information and Telecommunications March 22-26, 2009 TUNISIA

[12]. Matlab Image Processing Toolbox User's Guide version-2

[13]. Image processing study guide by Rafael C. Gonzalez,Richard E. Woods

[14]. Digital Image Processing By Gonzalez 2Nd Edition 2002, (Ebook) Prentice Hall.

[15]. Neural Network Toolbox™ 6 User's Guide by Howard Demuth , Mark Beale , Martin Hagan.