FPGA Implementation of Hybrid Architecture for Image Compression Optimized for Low Power and High Speed applications

1. Mr. Murali Mohan. S 2. Dr. P. Satyanarayana

Abstract: With the advancement in technology, many products in the market use images for control and display. Image compression is one of the primary image processing techniques that are embedded in all electronic products. In this paper, hybrid architecture comprising of DWT and Neural Network is combined together to compress and decompress image. The hybrid algorithm achieves 44% improvement in MSE and operates at 127MHz frequency on FPGA in compression and decompression of images. The software model developed in Matlab environment is validated using various test cases and the HDL model is simulated using XilinxISE/Modelsim. The design is synthesized and implemented on Virtex-5 FPGA and is optimized for area and power performances.

Key words: Discrete wavelet transforms(DWT), DWPT, FPGA, Hybrid architecture for Image compression, Hybrid technique, Neural networks.

1.Introduction

Image compression is one of the most promising subjects in image processing. Images captured need to be stored or transmitted over long distances. Raw image occupies memory and hence need to be compressed. With the demand for high quality video on mobile platforms there is a need to compress raw images and reproduce the images without any degradation. Several standards such as JPEG200, MPEG-2/4 recommend use of Discrete Wavelet Transforms (DWT) for image transformation which leads to compression with when encoded. Wavelets are a mathematical tool for hierarchically decomposing functions in multiple hierarchical sub bands with time scale resolutions. Image compression using Wavelet Transforms is a powerful method that is preferred by scientists to get the compressed images at higher compression ratios with higher PSNR values **[1].** It is a popular transform used for some of the image compression standards in lossy compression methods. Unlike the discrete cosine transform, the wavelet transform is not Fourier-based and therefore wavelets do a better job of handling discontinuities in data.

On the other hand, Artificial Neural Networks (ANN) **[2]** for image compression applications has marginally increased in recent years. Neural networks are inherent adaptive systems; they are suitable for handling nonstationaries in image data.

Artificial neural network can be employed with success to image compression. Image Compression Using Neural Networks by **Ivan Vilovic [3]** reveals a direct solution method for image compression using the neural networks. An experience of using multilayer perceptron for image compression is also presented. The multilayer perceptron is used for transform coding of the image. Image compression with neural networks by **J. Jiang [4]** presents an extensive survey on the development of neural networks for image compression which covers three categories: direct image compression by neural networks; neural network implementation of existing techniques, and neural network based technology which provide improvement over traditional algorithms. Neural Networks-based Image Compression System by **H. Nait Charif and Fathi. M. Salam [5]** describes a practical and effective image compression system based on multilayer neural networks. The system consists of two multilayer neural networks that compress the image in two stages. The algorithms and architectures reported in these papers sub divided the images into sub blocks and the sub blocks are reorganized for processing. Reordering of sub blocks leads to blocking artifacts. Hence it is required to avoid reorganization of sub blocks. One of the methods was to combine neural networks with wavelets for image compression **[6]**. Image compression using wavelet transform and a neural network was suggested previously **[7].** Wavelet networks (WNs) were introduced by **Zhang and Benveniste [8][9]** in 1992 as a combination of artificial neural networks and wavelet decomposition. Since then, however, WNs have received only little attention. In the wavelet networks, the basis radial functions in some RBF-networks are replaced by wavelets. **Szu et al. [10]** have shown usage of WNs for signals representation and classification. They have explained how a set of WN, "a super wavelet", can be produced and the original ideas presented can be used for the assortment of model. Besides, they have mentioned the big compression of data achieved by such a representation of WN's. **Zhang [11]** has proved that the WN's can manipulate the non-linear regression of the moderately big dimension of entry with the data of training. **Ramanaiah and Cyril [12]** in their paper have reported the use of neural networks and

• Mr.Murali Mohan.S, Associate Professor, Dept. of ECE, Sri Venkateswara College of Engineering & Technology, Chittoor, A.P., INDIA, Mobile No.: +91 9392547084,
e-mail ID: muralimohan.vlsi.dsp@gmail.com.
• Dr. P.Satyanarayana, Professor, Dept. of ECE, College of Engineering, S.V.University, Tirupati, A.P. , INDIA,
e-mail ID: satyamp1@yahoo.com.

ISSN 2229-5518

wavelets for image compression. In their work, the image is decomposed using DWT into four sub bands, and the neural network compresses the individual sub band and hence blocking artifacts error is minimized in the reconstructed image. Image decomposition using DWT into multiple sub bands leads to delay in compression, as the decomposition of image leads to multiple hierarchical sub blocks. In this paper we propose a novel approach for image compression using wavelets and neural networks. The input image is decomposed into four sub bands of LL, LH, HL and HH. Only the LL sub band is further decomposed in hierarchical sub bands until the sub band size is 8 x 8. The sub bands after decomposition using DWT are chosen based on information content and is further compressed using multilayered neural network architecture, thus minimizing the delay in compression.

Section II presents theoretical background on neural networks and DWT. Section III discusses the proposed image compression technique, section IV presents the implementation details and section V discuss the results and conclusion is presented in section VI.

2. Neural networks and DWT

In this section, neural network architecture for image compression is discussed. Feed forward neural network architecture and back propagation algorithm for training is presented. DWT based image transformation and compression is also presented in this section. Compression is one of the major subject of research, the need for compression is discussed as follows: Uncompressed video of size 640 x 480 resolution, with each pixel of 8 bit (1 bytes), with 24 fps occupies 307.2 Kbytes per image (frame) or 7.37 Mbytes per second or 442 Mbytes per minute or 26.5 Gbytes per hour. If the frame rate is increased from 24 fps to 30 fps, then for 640 x 480 resolutions, 24 bit (3 bytes) color, 30 fps occupies 921.6 Kbytes per image (frame) or 27.6 Mbytes per second or 1.66 Gbytes per minute or 99.5 Gbytes per hour. Given a 100 Gigabyte disk can store about 1-4 hours of high quality video, with channel data rate of 64Kbits/sec – 40 – 438 sec/per frame transmission. For HDTV with 720x1280 pixels/ frame, progressive scanning at 60 frames/ sec: 1.3Gb/s – with 20Mb/s available – 70% compression required – 0.35bpp. In this work we propose a novel architecture based on neural network and DWT [13].

2.1 Feed forward neural network architecture for image compression

An Artificial Neural Network (ANN) is an information- processing paradigm that is inspired by the way biological nervous systems, such as the Brian, process information. The key element of this paradigm is the novel structure of the information processing system. The basic architecture for image compression using neural network is shown in figure 1. The network has input layer, hidden layer and output layer. Inputs from the image are fed into the network, which are passed through the multi layered neural network. The input to the network is the original image and the output obtained is the reconstructed image. The output obtained at the hidden layer is the compressed image. The network is used for image compression by breaking it in two parts as shown in the Figure 1. The transmitter encodes and then transmits the output of the hidden layer (only 16 values as compared to the 64 values of the original image). The receiver receives and decodes the 16 hidden outputs and generates the 64 outputs. Since the network is implementing an identity map, the output at the receiver is an exact reconstruction of the original image.

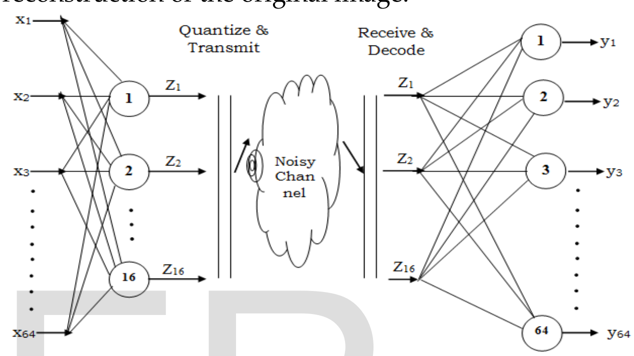


Figure 1 Feed forward multilayered neural network architecture

Three layers, one input layer, one output layer and one hidden layer, are designed. The input layer and output layer are fully connected to the hidden layer. Compression is achieved by designing the network such that the number of neurons at the hidden layer is less than that of neurons at both input and the output layers. The input image is split up into blocks or vectors of 8 X8, 4 X 4 or 16 X 16 pixels.

Back-propagation is one of the neural networks which are directly applied to image compression coding [14][15][16]. In the previous sections theory on the basic structure of the neuron was considered. The essence of the neural networks lies in the way the weights are updated. The updating of the weights is through a definite algorithm. In this paper Back Propagation (BP) algorithm is used to train feed forward network and to obtain optimum weights and biases for the hidden layer and the output layer. The algorithm is applied for the supervised learning that is a desired output will be applied to Neural Architecture [14]. Detailed discussion on back propagation algorithm is presented in [14][17].

2.2 DWT architecture for image compression

The DWT represents the signal in dynamic sub-band decomposition. Generation of the DWT in a wavelet packet allows sub-band analysis without the constraint of dynamic decomposition. The discrete wavelet packet transform (DWPT) performs an adaptive decomposition of frequency axis. The specific decomposition will be selected according to an optimization criterion. The Discrete Wavelet Transform (DWT), based on time-scale representation, provides efficient multi-resolution sub-band decomposition of signals. It has become a powerful tool for signal processing and finds numerous applications in various fields such as audio compression, pattern recognition, texture discrimination, computer graphics etc. Specifically the 2-D DWT and its counterpart 2- D Inverse DWT (IDWT) play a significant role in many image/video coding applications.

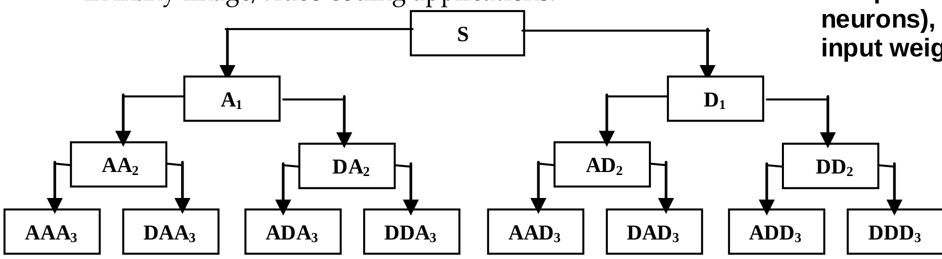


Figure 2 DWT decomposition

Figure 2 shows the DWT architecture, the input image is decomposed into high pass and low pass components using HPF and LPF filters giving rise to the first level of hierarchy. The process is continued until multiple hierarchies are obtained. A1 and D1 are the approximation and detail filters.

Figure 3 shows the decomposition results. The barbera image is first decomposed into four sub bands of LL, LH, HL and HH. Further the LL sub band is decomposed into four more sub bands as shown in the figure. The LL component has the maximum information content as shown in figure 3, the other higher order sub bands contain the edges in the vertical, horizontal and diagonal directions. An image of size N X N is decomposed to N/2 X N/2 of four sub bands. Choosing the LL sub band and rejecting the other sub bands at the first level compresses the image by 75%. Thus DWT assists in compression. Furhter encoding increases compression ratio.

(a) (b) (c)

Figure 3 DWT decomposition of barbera image (a) Input image (b) After one level 2D-DWT decomposition (c) After two level 2D-DWT decomposition

2.3 ANN with DWT for Image Compression

Basic architecture for image compression using neural network is shown in Figure 4. The input image of size 64 x 1 is multiplied by 4 x 64 weight matrixes to obtain the compressed output of 4 x 1, at the receiver 4 x 1 is decompressed to 64 x 1 by multiplying the compressed matrix by 64 x 4.

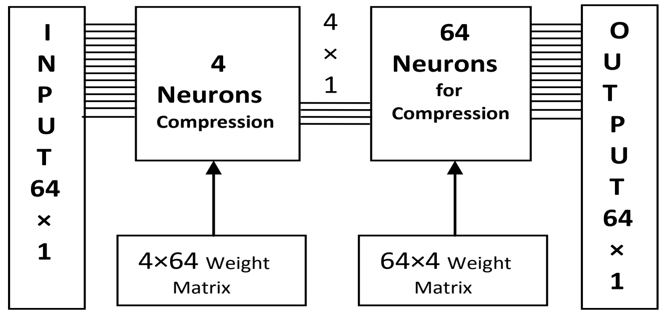


Figure 4 Neural network architecture for image compression with input layer (64x1), hidden layer (4 neurons), output layer (64 neurons), output layer(64x1), input weight (4x64) and output weight matrix (64x4).

Table 1 shows the compression ratio that can be achieved by choosing the sizes of hidden layer. If the input layer is set to 64, hidden layer to 4 and output layer to 64, the network compress 64 input pixels to 4 pixels at the hidden layer and is decompressed to 64 at the output layer. The compressed data is transmitted and at the receiver the output layer decompresses the received pixels to 64 pixels. The compression ratio is: (1-(4/64))*100% = 93.75%. By choosing the size of the hidden layer between 64 to 1, compression ratio between 0% to 98.5% can be achieved.

Table 1 Compression ratios for various neural network architectures

Network size	Size of hidden Layer	Compression Ratio
64-64-64	64	0%
64-32-64	32	50%
64-16-64	16	75%
64-08-64	08	87.5%
64-04-64	04	93.75%
64-01-64	01	98.5%

Prior to use of NN for compression it is required to perform training of the network, in this work back propagation training algorithm for obtaining the optimum weights and biases for the NN architecture is used. Set of test images have been chosen to form the training data ser. A test image such as barbera image is chosen to test the performance of the trained feed forward network. Figure 5 shows the original image which is the input image, compressed image and decoded image which is the decompressed image.

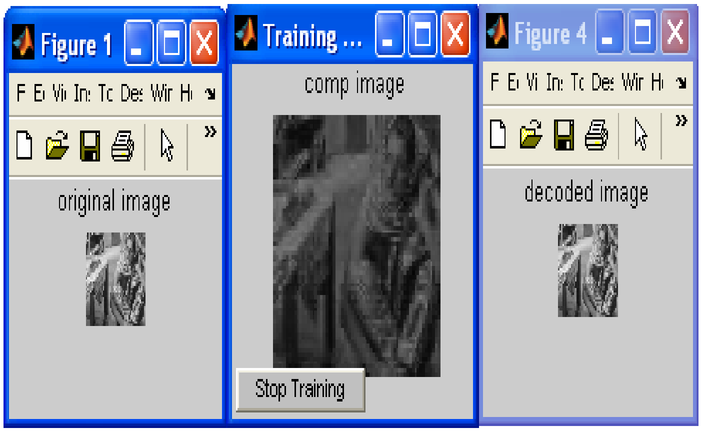


Figure 5 Test image results for barbera[17]

From the decompressed results shown in Figure 5 and Figure 6, it is found that the checker blocks error, which exists on the decompressed image. As the input image is sub divided into 8 x 8 blocks and rearranged to 64 x 1 input matrixes, the checker block arises. This is one of the limitations of NN based compression. Another major limitation is the maximum compression ration

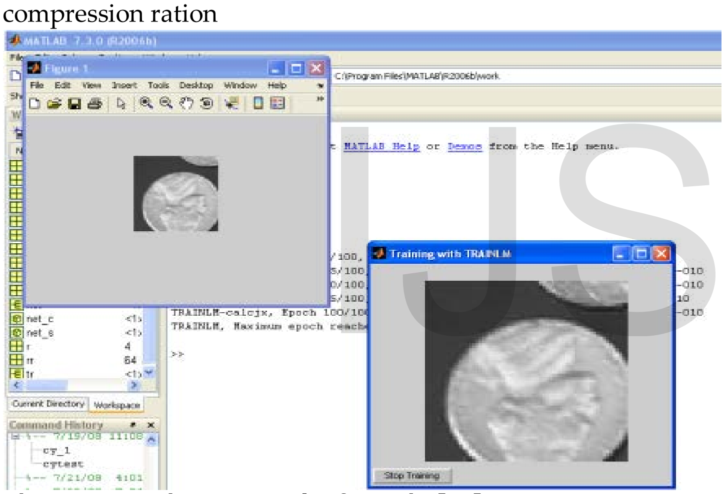


Figure 6 Test image results for coin [17]

which is less than 100%, in order to achieve compression more than 100% and to eliminate checker box errors or blocking artifacts we proposed DWT combined with NN architecture for image compression.

3. Image Compression

3.1 Proposed Technique for Image Compression

Most of the image compression techniques use either neural networks for compression or DWT (Discrete wavelet Transform) based transformation for compression.In order to overcome the limitations of NN architecture in this work, DWT is used for image decomposition and an N X N image is decomposed using DWT into hierarchical blocks the decomposition is carried out until the sub block is of size 8 x 8. For a image of size 64 x 64, first level decomposition gives rise to 32 x 32 (four sub bands) of sub blocks, further decomposition leads to 16 x 16 (sixteen sub bands), which can further decamped to 8 x 8 at the third hierarchy. The third level of hierarchy there are 64 sub blocks each of size 8 x 8. Figure 7 shows the decomposition levels of input image of size 64 x 64 using 2D-DWT. The sub bands are further decomposed into four sub bands as shown in Figure 7.

LL	LH	LL	LH	LL	LH	LL	LH
HL	HH	HL	HH	HL	HH	HL	HH
LL	LH	LL	LH	LL	LH	LL	LH
HL	HH	HL	HH	HL	HH	HL	HH
LL	LH	LL	LH	LL	LH	LL	LH
HL	HH	HL	HH	HL	HH	HL	HH
LL	LH	LL	LH	LL	LH	LL	LH
HL	HH	HL	HH	HL	HH	HL	HH

Figure 7 DWT Results after two level decomposition

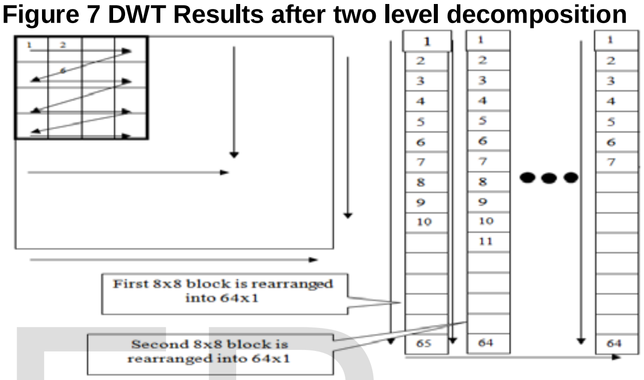


Figure 8 Decomposition of image into sub blocks using DWT

Figure 8 shows the rearrangement scheme adopted prior to compression using neural network architecture. Sub blocks of 8 x 8 are rearranged to 64 x 1 block are combined together into a rearranged matrix size as shown in figure 8. The rearranged matrix is used to train the NN architecture based on back propagation algorithm. In order to train the NN architecture and to obtain optimum weights it is required to select appropriate images. The training vectors play a vital role in NN architecture for image compression. Figure 9 shows the training sets for NN architecture.

Figure 9 Training set for NN architecture

The NN architecture consisting of input layer, hidden layer and output layer. The hidden layer consists of network function of four types shown in Table 1. Similarly the output layer also can be any of the four network functions. It is required to choose appropriate network function.

Table 2 Neural network classification based on transfer function

Neural network type	Transfer function	
	Hidden layer	Output layer
Linear network	Purelin	Purelin
Nonlinear network	Tansig or logsig	Tansig or logsig
Hybrid network	Tansig or logsig	Purelin

Trainrp is a network training function used in this work that updates weight and bias values according to the resilient back propagation algorithm (RP). Trainlm may also be used which is also a network training function that updates weight and bias values according to Levenberg algorithm but consumes more memory. Based on the above parameters chosen the Hybrid Compression Algorithm is developed and is shown in Figure 10.

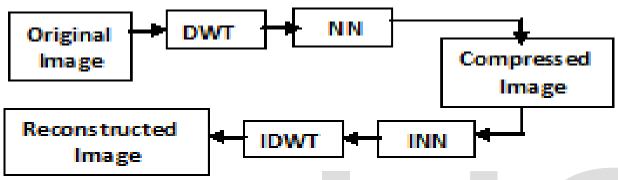


Figure 10 Proposed hybrid algorithms for image compression

3.2 Results and Discussion

Training the network using the test training sets and selection of appropriate network functions, Matlab model is developed and is used for analysis. Table 3 and Table 4 show the results for Pears image for various network functions and hidden layer sizes.

Table 3 NN performance for various network functions for Pears

Pears	MSE-Pears				
Net-work	Purelin-Purelin	Tansig - Tansig	Tansig - Purelin	Purelin - Tansig	Logsig - Logsig
[8 64]	26.776	6.23E+02	53.5689	4.65E+02	94
[16 64]	16.0131	1.76E+02	37.423	5.33E+02	4.53E+02
[32 64]	14.623	71.8136	39.897	58.8547	3.18E+02
[40 64]	15.888	62.1854	40.607	56.2962	1.14E+02

Table 4 NN performance for various network functions for Trees

Pears	MSE-Trees				
Net-work	Purelin-Purelin	Tansig-Tansig	Tansig-Purelin	Purelin - Tansig	Logsig-Logsig
[8 64]	7.577	120.4427	1.53E-08	84.2421	189.8105
[16 64]	11.1768	58.6797	0.7412	74.2862	1.15E+03
[32 64]	20.5682	32.5422	29.6525	56.2924	82.2705
[40 64]	11.0607	25.6774	44.8634	46.4044	70.5517

From the results presented in Table 3 and Table 4 shows that for NN architecture of [8 64] (8 neurons in hidden layer and 64 in the output layer), MSE is very less for Tansig-Purelin. Hence in this work, we propose tansig and purelin are the network functions for NN architecture and are called as Hybrid NN architecture. Figure 11 shows the MSE, PSNR and Max Error parameters for various input block size.

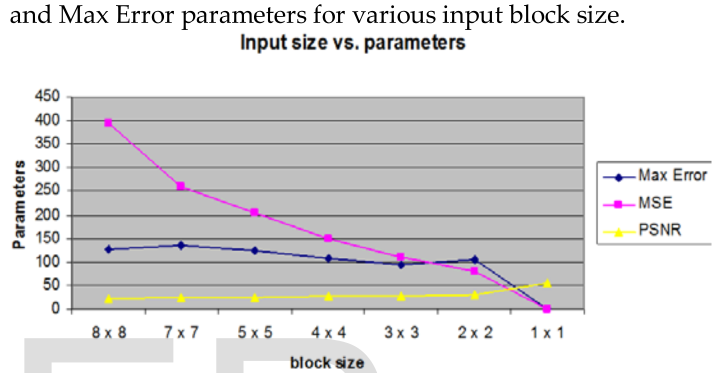


Figure 11 Input block size and NN performance

From the results shown in figure 11 it is found that lesser input layer size better is MSE, however if the input layer size less it also increases the complexity of NN architecture. Number of hierarchical levels in DWT need to be increased, hence in this work we choose 8 x 8 block size, the input image is divided into 8 x 8 block size using DWT. Figure 12 shows the results of selection of number of hidden layers. The input layer consisting of 64 x 1 can be compressed to 16 x 1, which can be further compressed to 8 x 1, and further to 4 x 1 and can be reconstructed to 64 x 1 at the output layer. The results shown are analyzed for three images, from the results it is found that increasing the number of hidden layers does not improve NN compression performance. Hence the network chosen in this work consists of input layer of 64 x 1, hidden layer of 4 x 1 and output layer of 64 x 1. The network functions are tansig in the hidden layer and purelin in the output layer.

In this proposed architecture, the input image is first decomposed into multiple sub blocks using hierarchical DWT architecture, the decomposed image is reordered and is processed using the NN architecture. The NN architecture compresses the transformed image, appropriate weights and

ISSN 2229-5518

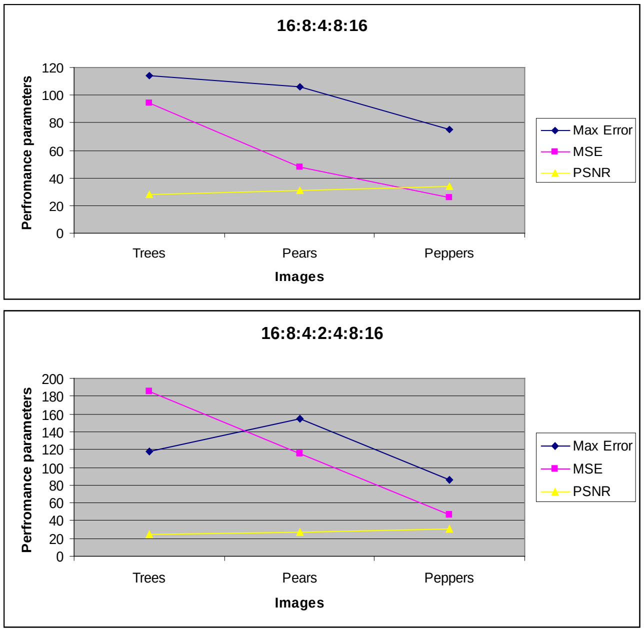
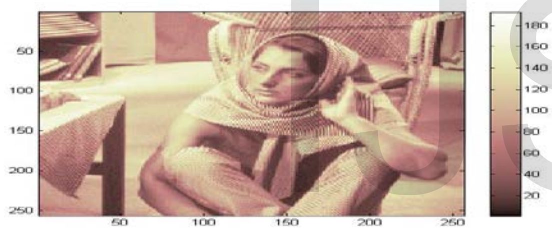
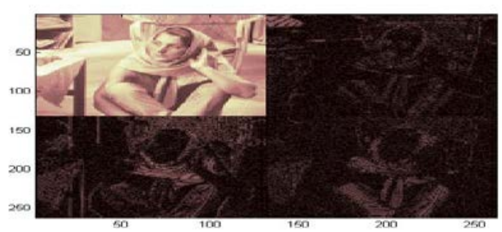


Figure 12 NN performances for various hidden layers

biases are chosen for compression and decompression. Hybrid network functions are used for NN architecture. The decompressed image is reconstructed using IDWT.

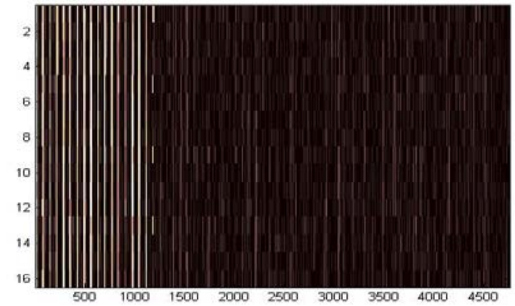


(a)

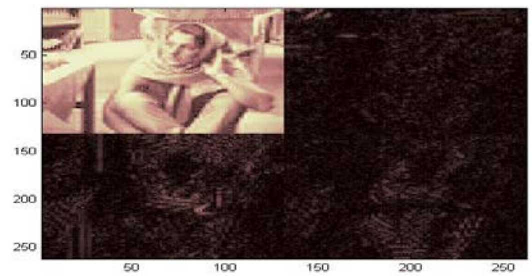


(b)

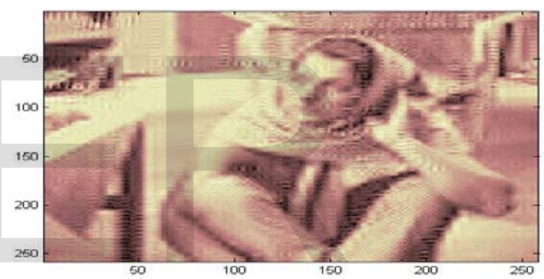
(c)



(d)



(e)



(f)

Figure 13 Image Compression and Decompression using Hybrid Technique

The input image is decomposed into sub bands using DWT, the input image of size N x N is decomposed into 1-level (Shown in Fig. 13(b)), further the four sub bands is decomposed into 8 sub bands using DWT (Shown in Fig. 13(c)), the decomposed sub bands are reorganized into column matrix (shown in Fig. 13 (d)), the neural network architecture compresses the reorganized image and the compressed image is further decompressed using the output layer(compressed output and decompressed output are not shown in the figure). The decompressed output from the output layer is rearranged and inverse DWT is performed (shown in Fig. 13 (e)), the second level inverse DWT is performed and the final output is reconstructed to obtain the original image (shown in Fig. 13(f)). From the reconstructed image it is found that the checker box errors that were existing using neural networks have been reduced and the results of MSE is shown in Table 5. The results are compared with the results of image compression using NN.

Table 5 Results comparison

Test Image	Image MSE (With NN only)	Image MSE (With NN and DWT)
cell	39	26
circuit	24	14
Lena	201	100
girl	67	42
Blue hills	38	22
Sunset	51	39
Water lilies	56	31
Winter	89	47
College	180	97
Garden	163	87
Bhagavadgeetha	98	65
Devinecouple	143	80

4. Implementation

4.1 FPGA Implementation of DWT-NN architecture

The hybrid architecture proposed in this work consists of two level 2-D DWT/IDWT and Neural network hidden layer and output layer. The input image of size 64x64 is first decomposed to four sub bands of 32x32 in the first level decomposition and to further 8 sub bands of 16x16 in the second level. The 16x16 image sub bands are grouped into 8x8 blocks, thus each 16x16 sub band consists of four 8x8 sub images, for the two level decomposed images there are 32 sub images. Each sub image is reorganized to 64x1, thus the decomposed image is reorganized to 64x32 input arrays. The input array is stored in intermediate memory. The input layer of neural network architecture is of 64x1 is compressed by the hidden layer consisting of 4 neurons, the decompressor or the output layer consisting of purelin network function decompressor the 4 neurons to 64 neurons. Two level 2D-IDWT reconstructs the output of neural network decompressor to the reconstructed image. The computation complexity of the proposed architecture is complex as there are several data processing modules and intermediate memory. The proposed architecture is implemented on FPGA.

4.2 DWT Architecture and Neural Network Architecture

Figure 14 shows the 2D DWT architecture using lifting scheme logic. Lifting-based DWT requires less computation compared to the convolution-based approach. However, the savings depend on the length of the filters. During the lifting implementation, no extra memory buffer is required because of the in place computation feature of lifting. This is particularly suitable for the hardware implementation with limited available on-chip memory. The basic principle of the lifting scheme is to factorize the polyphase matrix of a wavelet filter into a sequence of alternating upper and lower triangular matrices and a diagonal matrix.

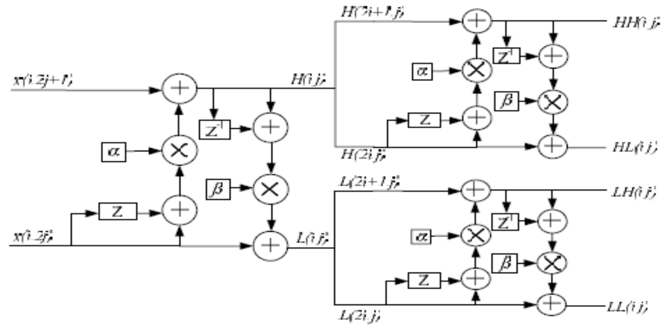


Figure 14 5/3 1-D lifting-based DWT

The 5/3 2-D DWT is a multilevel decomposition technique that decomposes into four sub bands such as HH, HL, LH and LL. The mathematical formulas of 5/3 2-D DWT is defined as follows:

$$HH(i, j) = H(2i+1, j) + \alpha\,(H(2i, j) + H(2i+2, j))$$
$$HL(i, j) = H(2i, j) + \beta\,(HH(i, j) + HH(i-1, j))$$
$$LH(i, j) = L(2i+1, j) + \alpha\,(L(2i, j) + L(2i+2, j))$$
$$LL(i, j) = L(2i, j) + \beta\,(LH(i, j) + LH(i-1, j))$$

In this architecture of DWT, the data flow of the horizontal filter and the vertical filter can be derived. The architecture performs in (4N 2 (1 − 4 j) + 9N) / 6 computation time, when the size of the input image is $N \times N$ and the number of compression level is j. The architecture of 5/3 2-D DWT involves four processor elements, 3.5N+8 registers and seven multiplexers. The hardware utilization is 100%. Verilog models have been developed for the 2D DWT architecture and the HDL models have been simulated.

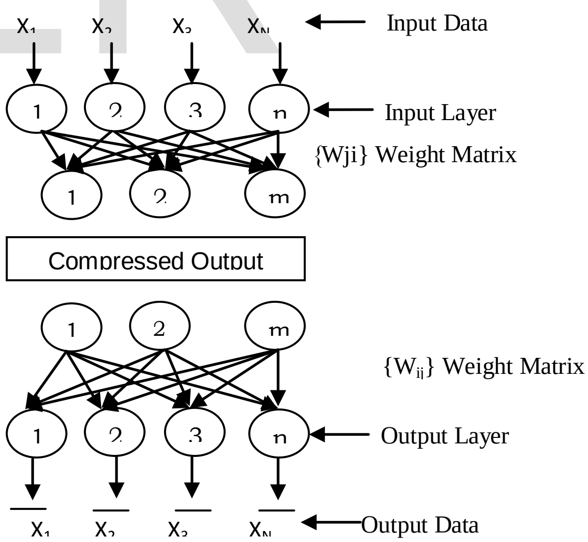


Figure 15 Neural network compressor and decompressor

The simulated design is synthesized and the reports generated for analysis Figure 15 shows the neural network architecture for compression of rearranged sub bands from DWT output. The input of size 64x1 is compressed to 4 x1 using the hidden layer and is reconstructed to 64x1 using the output layer. The network consists of multipliers and adders for weight multiplication and accumulation.

5. Results & Conclusion

5.1 Simulation and Synthesis Results

The simulation results of hybrid architecture with 64 inputs have been found to be compressed to 4 outputs, and further decompressed to 64 outputs at the output layer. Known sets of input vectors have been used to validate the results. Table 6 shows the device utilization report on Virtex-5 FPGA, from the synthesis report we find that the proposed architecture utilizes only 6% of the total slices and utilizes 3% of DSP blocks.

Table 6 FPGA resource utilization results

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	4,463	69,120	6%
Number of Slice LUTs	5,323	69,120	7%
Number used as logic	4,244	69,120	6%
Number used as Memory	1,009	17,920	5%
Number of occupied Slices	2,103	17,280	12%
Number of LUT Flip Flop pairs used	7,275		
Number with an unused Flip Flop	2,812	7,275	38%
Number with an unused LUT	1,952	7,275	26%
Number of fully used LUT-FF pairs	2,511	7,275	34%
Number of slice register sites lost to control set restrictions	133	69,120	1%
Number of bonded IOBs	2	640	1%
Number of LOCed IOBs	2	2	100%
Number of Block RAM/FIFO	19	148	12%
Total Memory used (KB)	666	5,328	12%
Number of BUFG/BUFGCTRLs	2	32	6%
Number of BSCANs	1	4	25%
Number of DSP48Es	2	64	3%

Table 7 shows the comparison of the synthesis report on implementation using Spartan III and Virtex 5 FPGA. The design occupies equal number of slices on both the devices; however on Virtex-5 FPGA the utilization factor is very low as they have more number of devices. The timing report illustrates the maximum frequency of operation is 127MHz and consumes power of less than 0.45mW on Virtex-5 FPGA.

Table 7 Synthesis Report Comparison

Device/Logic	Spartan 3 Xc3s50-5tq144	Virtex 4 Xc4vlx15-12ff676
Total Number Slice Registers available	1536	12,288
Total Number Slice Registers used	733	733
Utilization of Slice registers	47%	5%
Number 0f used Flip Flops	361	361
Number of used as Latches	372	372
Number of 4 input LUTs available	1536	12,288
Number of 4 input LUTs used	806	806
Utilization of 4 input LUTs	52%	6%
Number of occupied Slices available	768	6,144
Number of occupied Slices used	593	593
Utilization occupied slices	77%	9%
Utilization only related logic	100%	100%

Figure 16 shows the technology schematic of the hybrid architecture, the technology schematic is obtained using Xilinx ISE 13.1 version.

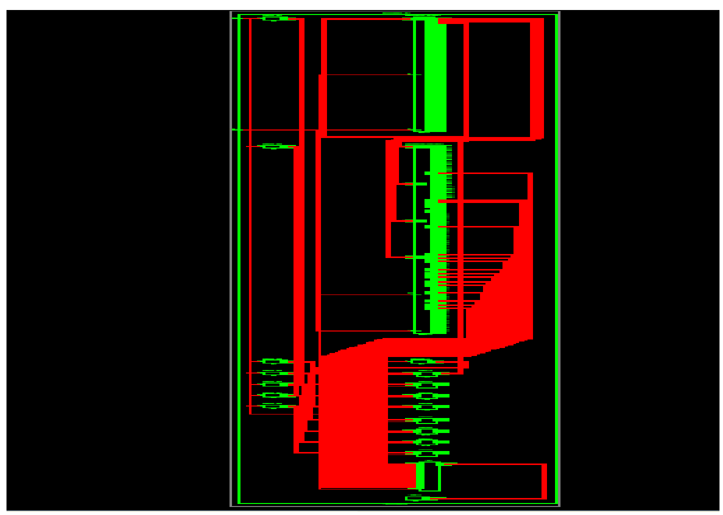


Figure 16 RTL schematic of Hybrid Architecture

5.2 Conclusion

In this paper the methods for optimal computer aided designs of selected Hybrid NN based image compression techniques are carried out. The design is simulated in

ISSN 2229-5518

MATLAB and the software reference model is verified. The weights and biases obtained using the software environment is used in developing HDL model. Multiplier and adder modules are developed and FSM is used to control the data flow. The RTL code is synthesized using Xilinx ISE and is targeted on Virtex-5 FPGA. The hybrid architecture for image compression is verified using various test images and the MSE is found to be better compared with Neural Network model. The synthesized design is optimized for are and power performances. The computation complexity of the hybrid architecture can be further reduced by designing the subsystems, selection of appropriate data path operators and state machines for data flow logic.

Acknowledgement: The authors would like to thank Dr. Cyril Prasanna Raj P., for his valuable support and inputs provided in carrying out his work.

References:

[1] Q. Zang, Wavelet Network in Nonparametric Estimation. IEEE Trans. Neural Networks, 8(2):227- 236, 1997

[2] Q. Zang and A. Benveniste, Wavelet networks. IEEE Trans. Neural Networks, vol. 3, pp. 889-898, 1992.

[3] A. Grossmann and B. Torrésani, Les ondelettes, Encyclopedia Universalis, 1998.

[4] R. Ben Abdennour, M. Ltaïef and M. Ksouri. uncoefficient d'apprentissage flou pour les réseaux deneurones artificiels, Journal Européen des Systèmes Automatisés, Janvier 2002.

[5] M. Chtourou. Les réseaux de neurones, Support de cours DEA A-II, Année Universitaire 2002/2003.

[6] Y. Oussar. Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus, Thèse de doctorat, Université Pierre et Marie Curie, juillet 1998.

[7] R. Baron. Contribution à l'étude des réseaux d'ondelettes, Thèse de doctorat, Ecole Normale
Supérieure de Lyon, Février 1997.

[8] C. Foucher and G. Vaucher. Compression d'images et réseaux de neurones, revue Valgo n°01-02, 17-19 octobre 2001, Ardèche.

[9] J. Jiang. Image compressing with neural networks – A survey, Signal processing: Image communication, ELSEVIER, vol. 14, n°9, 1999, pp. 737-760.

[10] S. Kulkarni, B. Verma and M. Blumenstein. Image Compression Using a Direct Solution Method Based Neural Network, The Tenth Australian Joint Conference on Artificial Intelligence, Perth, Australia, 1997, pp. 114-119.

[11] G. Lekutai. Adaptive Self-tuning Neuro Wavelet Network Controllers, Thèse de Doctorat, Blacksburg-Virgina, Mars 1997.

[12] R.D. Dony and S. Haykin. Neural network approaches to image compression, Proceedings of the IEEE, V83, N°2, Février, 1995, pp. 288-303.

[13] A. D'souza Winston and Tim Spracklen. Application of Artificial Neural Networks for real
time Data Compression, 8th International Conference On Neural Processing, Shanghai, Chine, 14-18 Novembre 2001.

[14] Ch. Bernard, S. Mallat and J-J Slotine. Wavelet Interpolation Networks, International Workshop on CAGD and wavelet methods for Reconstructing Functions, Montecatini, pp.15-17 Juin 1998.

[15] D. Charalampidis. Novel Adaptive Image Compression, Workshop on Information and Systems Technology, Room 101, TRAC Building, University of New Orleans, 16 Mai 2003.

[16] M. J. Nadenau, J. Reichel, and M. Kunt, "Wavelet Based Color Image Compression: Exploiting the Contrast Sensitivity Function", *IEEE Transactions Image Processing*, vol. 12, no.1, 2003, pp. 58-70.

[17] K. Ratakonda and N. Ahuja, "Lossless Image Compression with Multiscale Segmentation", *IEEE Transactions Image Processing*, vol. 11, no.11, 2002, pp. 1228-1237.

[18] K. H. Talukder and K. Harada, "Haar Wavelet Based Approach for Image Compression and
Quality Assessment of Compressed Image", *IAENG International Journal of Applied Mathematics*, 2007.

[19] Bo-Luen Lai and Long-Wen Chang, "Adaptive Data Hiding for Images Based on Haar Discrete Wavelet Transform", *Lecture Notes in Computer Science*, Springer-Verlag, vol. 4319, 2006, pp. 1085-1093.

[20] S. Minasyan, J. Astola and D. Guevorkian, "An Image Compression Scheme Based on Parametric Haar-like Transform", *ISCAS 2005. IEEE International Symposium on Circuits and Systems*, 2005, pp. 2088-2091.

[21] Z. Ye, H. Mohamadian and Y.Ye, "Information Measures for Biometric Identification via 2D Discrete Wavelet Transform", *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, CASE'2007*, 2007, pp. 835-840.

[21] S. Osowski, R. Waszczuk, P. Bojarczak, "Image compression using feed forward neural networks — Hierarchical approach" *Lecture Notes in Computer Science*, Book Chapter, Springer-Verlag, vol. 3497, 2006, pp. 1009- 1015.

[22] M. Liying and K. Khashayar, "Adaptive Constructive Neural Networks Using Hermite Polynomials for Image Compression", *Lecture Notes in Computer Science*, Springer-Verlag, vol. 3497, 2005, pp. 713-722.

[12] R. Cierniak, "Image Compression Algorithm Based on Soft Computing Techniques",*Lecture Notes in Computer Science*, Springer-Verlag, vol. 3019, 2004, pp. 609-617.

[13] B. Northan, and R.D. Dony, "Image Compression with a multiresolution neural network", *Canadian Journal of Electrical and Computer Engineering*, Vol. 31, No. 1, 2006, pp. 49-58.

[14] S. Veisi and M. Jamzad, "Image Compression with Neural Networks Using Complexity Level
of Images", *Proceedings of the 5^{th} International Symposium on image and Signal Processing and Analysis, ISPA07*, IEEE, 2007, pp. 282-287.

[15] I. Vilovic, "An Experience in Image Compression Using Neural Networks", *48^{th} International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications*, IEEE, 2006, pp. 95-98.

[16] J. Mi, D. Huang, "Image Compression using Principal Component Neural Network", *8th
International Conference on Control, Automation, Robotics and Vision*, IEEE, 2004, pp. 698-701.

[17] R. Ashraf and M. Akbar, "Absolutely lossless compression of medical images", *27th Annual Conference Proceedings of the 2005 IEEE Engineering in Medicine and Biology*, IEEE, 2005, pp. 4006-4009.

[18] A. Khashman and K. Dimililer, "Neural Networks Arbitration for Optimum DCT Image Compression", Proceeding *of the IEEE International Conference on 'Computer as a Tool' EUROCON'07*, 2007, pp. 151-156.

[19] A. Khashman and K. Dimililer, "Intelligent System for Image Compression", Proceeding of 9th International Conference on Enterprise Information Systems, ICEIS 2007, 2007, pp. 451-454.

[20] A. Khashman and K. Dimililer, "Comparison Criteria for Optimum Image Compression", *Proceeding of the IEEE International Conference on 'Computer as a Tool' EUROCON'05*, vol. 2, 2005, pp. 935-938.

[21] A. Khashman and K. Dimililer, "Haar Image Compression Using a Neural Network", *Proceedings of the WSEAS Int. Applied Computing Conference (ACC'08)*, Istanbul, Turkey, 27-29 May 2008.

[22] A. Khashman, B. Sekeroglu, and K. Dimililer, "Intelligent Identification System for Deformed Banknotes", *WSEAS Transactions on Signal Processing, ISSN 1790-5022, Issue 3, Vol. 1*, 2005.