DOI: 10.21917/ijme.2021.0183

FPGA IMPLEMENTATION OF CSD BASED NN IMAGE COMPRESSION ARCHITECTURE

M. Lakshmi Kiran, K. Nikhileswar and K. Venkata Ramanaiah

[1]*Department of Electronics and Communication Engineering, PVKK Institute of Technology, India*
[2]*Department of Electrical Engineering, Indian Institute of Technology, Patna, India*
[3]*Department of Electronics and Communication Engineering, Yogi Vemana University, India*

Abstract

Complexity will be the critical issue in Very Large Scale Integration (VLSI) implementation of Image Compression Architectures. Especially it will be predominant issue while dealing with Neural Network (NN) based image compression architectures. Due to the development of Field-Programmable Gated Array (FPGA), dealing of NN Image Compression Architecture becomes smoother. Furthermore reducing power consumption of those architectures can be deal with Canonic Signed Digit (CSD) algorithms. NN based compression can be added to standard JPEG compression is proved be an efficient strategy for dealing images of high resolution in terms of speed and power. CSD algorithm is proved to provide low power consumption along with low computation time while dealing NN structures. The proposed architecture is hence based on CSD Floating Point Matrix Multiplier (FPMM) and it is synthesized and implemented using Xilinx Vivado Artix7 and Nexys DDR boards. Simulation with MATLAB & Xilinx Vivado is carried out for this work and almost same results are observed.

Keywords:

CSD, FPMM, DDR, FPGA

1. INTRODUCTION

Image compression is one type of data compression, without which video conferencing, video streaming, video calling and such multimedia related applications that are available to common users can't be imagined. Number of standard algorithms is available in the literature for image compression such as JPEG, MPEG and etc.

Most of the times hardware implementation [1] enables improvements in some parameters like peed of operation against that of software implementation. Architectures of image compression may also come under this category. FPGA is one such a beautiful reconfigurable hardware platform, in which prototype can be developed for image compression. Day by day number of components in FPGA are getting more and more, and it enables to do more amount of image processing at a time.

Fortunately internal structure of FPGA is very much suitable for implementing Neural Network (NN) based architectures [3], which allows parallel processing, adoptability and etc. Moreover image compression done with NNs which is unlike conventional image compression methods. Hence NN based image compression can be efficiently implemented on FPGA. But, NN structure contains many number of adder and multiplier elements. Due to this power consumption can be larger. This can be tackled using CSD algorithm [2], as it encodes the data with minimum number of non-zeros and hence minimum number of multiplications only required. Thus power consumption reduction is possible to a great extent along with the improvement in the speed of computation.

The section 2 presents internal structure of NN based image compression, Section 3 covers hardware realization of NN image compression, section 4 contains Matlab simulation results, Xilinx Synthesis and implementation results for hardware realization of NN image compression and conclusion and future scope for this work is available in section 5.

2. NN BASED IMAGE COMPRESSION

Image compression with Neural Network is as shown in the Fig.1, containing 3 types of layers, namely input layer, hidden layer and output layers. Generally neurons in input layer and output layer are equal and greater than that in hidden layer. Image after some pre-processing steps results to row of values, which are applied at input side, output of hidden layer corresponds to compressed image as it is encoded with lesser number of neurons than at input. Output of output layer is corresponds to reconstructed image.

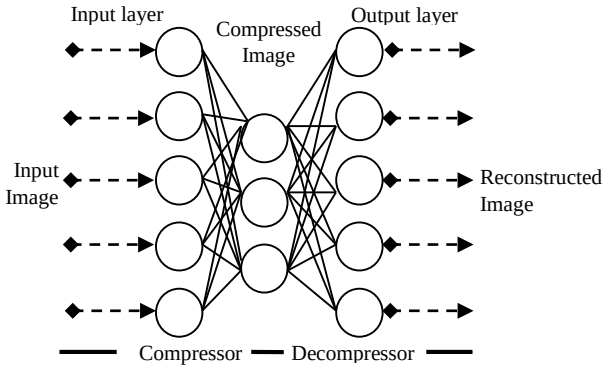


Fig.1. Compression stage with NN architecture

3. NN BASED IMAGE COMPRESSION

This section explains the hardware realization of NN Image Compression, which is explained at earlier section.

First of all, given image is trained with the neural network in many number of times until it leads to have least error. Once it is trained, optimized weight matrix and bias matrix along with the input image matrix is considered in hardware realization as observed in Fig.2. It mainly containing multiplier and adder. Hardware realization of NN image compression requires many number of multipliers and adders, while dealing in real time scenario, although a small image or sub-image is considered here form discussion. To make better design of this, multiplier should

be designed as efficiently as possible i.e. optimized power, area and speed.

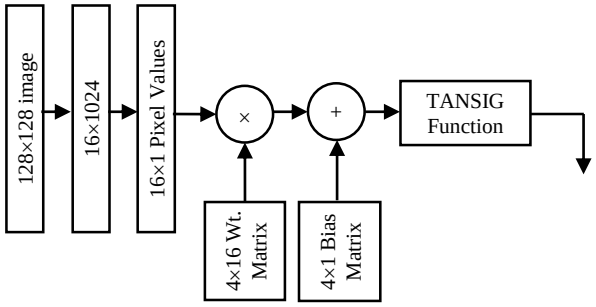
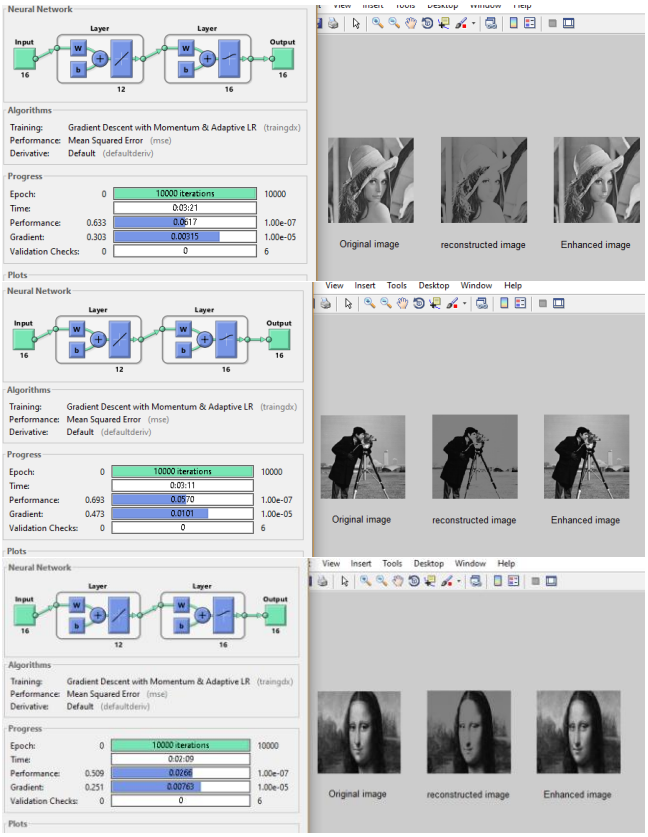


Fig.2. Block Diagram for Hardware Realization of NN compression

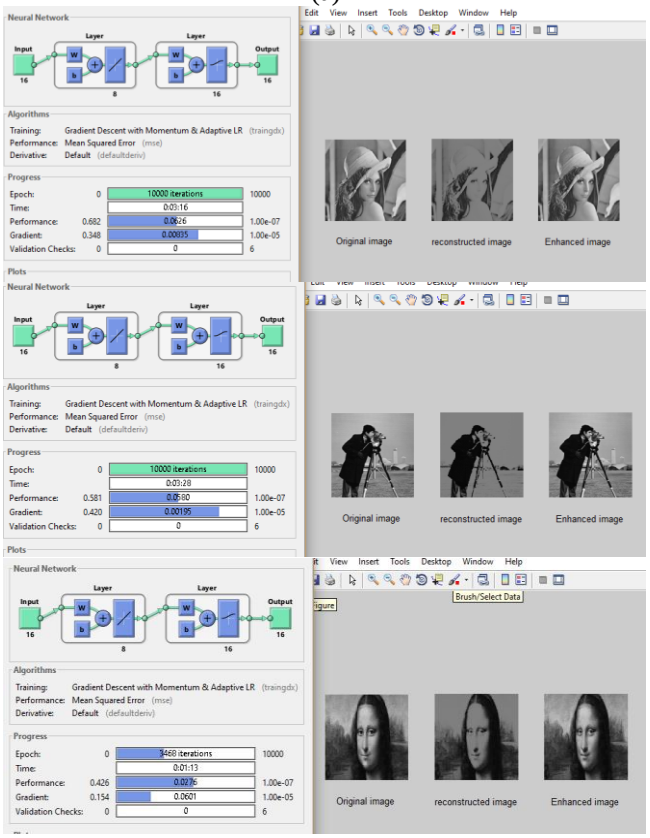
Canonic Signed Digit (CSD) algorithm [4] is such an algorithm, which converts binary numbered input into CSD numbered (-1, 0 and 1 for Radix-2 case) output. Due to this the number of non-zeros will be reduced so that number multiplications in a product of two numbers can be decreased. Thus, it reduces power consumption and improves speed of computation but at the cost of complexity. However this complexity of the design is not a big deal, when implementing this design with the FPGA.

4. SYNTHESIS AND IMPLEMENTATION

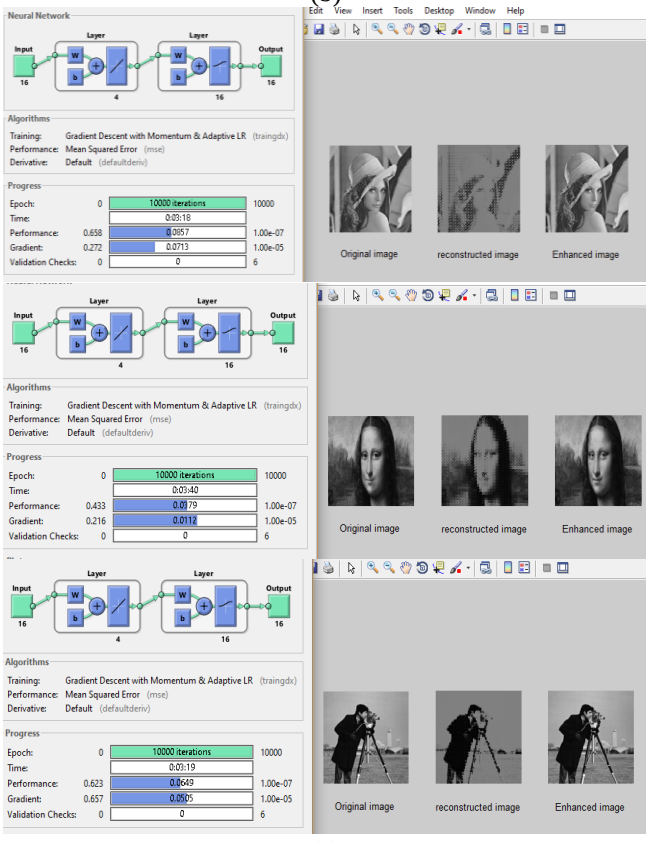
Standard images like Lenna, Camerman and Monalisa are taken into consideration for training them by using MATLAB NNTOOL. Number of neurons at different stages is selected in a way to produce 25%, 50% and 75% compression rate as shown in Fig.3.



(a)



(b)



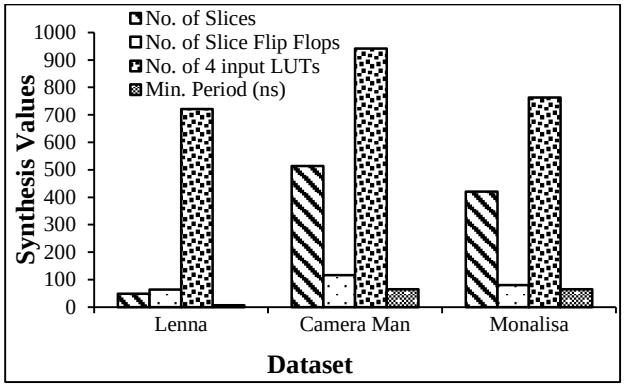
(c)

Fig.3. Lenna Image, Cameraman Image and Monalisa Images Compression and Decompression at (a) 25% (b) 50% and (c) 75%

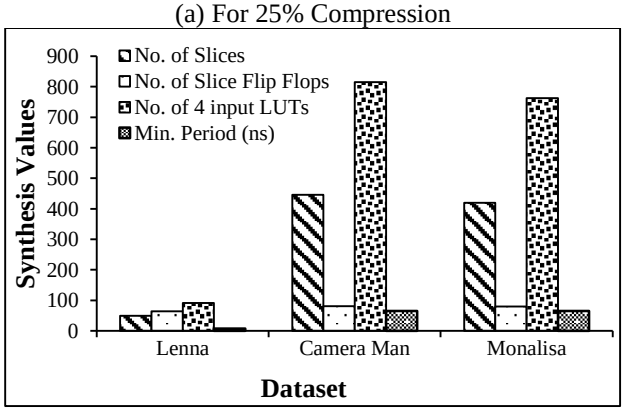
Table.1. Synthesis results of Matrix multiplication

Images	Sub Image Size	Weight Matrix Size	Number of Slices (Out of 7680)	Number of Slice Flip Flops (Out of 15360)	Number of 4 input LUTs (Out of 15360)	Maximum output required time after clock (ns)	Minimum period (ns)
1 (Lenna)	16×1024	12×16	49	64	721	7.447	6.951
2 (Cameraman)	16×1024	12×16	514	116	941	7.447	65.373
3 (Monalisa)	16×1024	12×16	420	80	763	7.447	64.768
4 (Lenna)	16×1024	8×16	49	64	91	7.447	6.951
5 (Cameraman)	16×1024	8×16	446	81	815	7.447	64.768
6 (Monalisa)	16×1024	8×16	420	80	763	7.447	64.768
7 (Lenna)	16×1024	4×16	49	64	721	7.447	6.951
8(Cameraman)	16×1024	4×16	400	80	812	7.447	65.768
9 (Monalisa)	16×1024	4×16	415	70	750	7.447	64.768

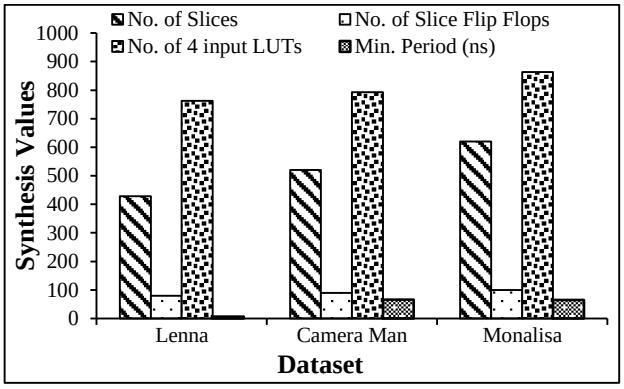
However compression rate increased from 25% to 50% and then 75%, there is small change that is observed in the quality of the images. It is observed that MATLAB results are exactly matched with the Xilinix results for multiplication of Image matrix with the Weight matrix [5]. Synthesis of this design is done while targeting on Artix-7 of Xilinx Vivado, and observations of it are tabulated below. Number of slices, minimum time required for computation w.r.t size is observed in the synthesis of it. However it is observed that maximum required time for various images is constant irrespective of its rate of compression. This will be greatly useful while working with real time image compression applications of variable sized images. Synthesis results of Matrix multiplication for NN image compression as in Table.1.



(a) For 25% Compression



(b) For 50% Compression



(c) For 75% Compression

Fig.4: Comparison of synthesis values of standard images with (a) 25% (b) 50% and (c) 75% compression rates

The Fig.4 represents comparison of synthesis of 3 standard images in three different compression rates. The design was implemented by targeting on Nexys Board and Zed Board (Zynq) through Xilinx Vivado Tool. Nexys 4 DDR board is a ready to use digital circuit development platform with Artix 7 FPGA developed by Xilinx. The Nexys Board results of Post synthesis and Post implementation of Matrix multiplier are shown in Table.2 and Table.3 respectively.

Table.2. Post Synthesis

Resource	Utilization	Available	% Utilization
LUT	479	53200	0.75
FF	112	126800	0.08
IO	1	210	0.47
BUFG	2	32	6.25

Table.3. Post implementation

Resource	Utilization	Available	% Utilization
LUT	959	63400	1.51
LUTRAM	24	19000	0.12

FF	1010	126800	0.79
IO	1	210	0.47
BUFG	3	32	9.3

Zedboard is also a total development kit for designers interested to execute designs through Xilinx Zynq 7000. Matrix multiplication algorithm is implemented and its post synthesis and Post implementation results are as shown in Table.4 and Table.5 respectively.

Table.4. Post Synthesis

Resource	Utilization	Available	% Utilization
LUT	479	53200	0.90
FF	112	106400	0.10
IO	1	200	0.5
BUFG	2	32	6.25

Table.5. Post implementation

Resource	Utilization	Available	% Utilization
LUT	957	53200	1.79
LUTRAM	24	17400	0.13
FF	1010	106400	0.94
IO	1	200	0.5
BUFG	3	32	9.37

It is observed that both Zedboard and Nexys 4 DDR give equal amount of delay in terms of WNS, WHS and WPWS, but Zedboard is power efficient and consumes power around 101mW whereas Nexys4 DDR board consumes 126mW as shown in Fig.5.

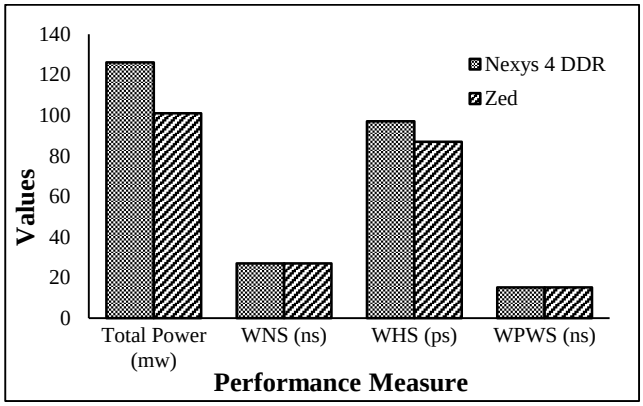


Fig.5. Comparison of Nexys 4 DDR and Zedboard

5. CONCLUSION

Design towards finding Hardware algorithms for image compression and its FPGA implementation is done in this paper. NN based compression is proved as better in comparison with the existing standard methods, and CSD algorithm for multiplier design is used for making effective VLSI design for NN image compression architecture. But, it has done only for Gray Scale images. It needs to be applied for Color images and complexity caused by CSD algorithm is to be reduced further. ASIC implementation of the proposed design is also be done for accurate estimation in real time fabrication.

REFERENCES

[1] Saudagar Abdul Khader Jilani and Syed Abdul Sattar, "JPEG Image Compression using FPGA with Artificial Neural Networks", *International Journal of Engineering Technology*, Vol. 2, No. 3, pp. 252-257, 2010.

[2] Sampatrao L. Pinjare and E. Hemanth Kumar, "Implementation of Artificial Neural Network Architecture for Image Compression Using CSD Multiplier", *Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications*, pp. 1-13, 2019.

[3] Kaiyuan Guo, Shulin Zeng, Jincheng Yu, Yu Wang and Huazhong Yang, "A Survey of FPGA-Based Neural Network Inference Accelerator", *ACM Transactions on Intelligent Systems and Technology*, Vol. 9, No. 4, pp. 1-26, 2017.

[4] Keshab K. Parhi, "*VLSI Digital Signal Processing Systems: Design and Implementation*", Wiley, 1999.

[5] Lakshmi Kiran Mukkara and K. Venkata Ramanaiah, "CSD Based VLSI Architecture for NN Based Image Compression", *International Journal of Recent Technology and Engineering*, Vol. 7, pp. 50-53, 2019.

[6] E.D. Kusuma and T.S. Widodo, "FPGA Implementation of Pipelined 2D-DCT and Quantization Architecture for JPEG Image Compression", *Proceedings of International Symposium on Information Technology*, Vol. 1, pp. 1-6, 2010.

[7] A. Sindhuja and V. Sadasivam, "Wavelet based Segmentation using Optimal Statistical Features on Breast Images", *ICTACT Journal on Image and Video Processing*, Vol. 4, No. 4, pp. 853-857, 2014.

[8] T. Pradeepthi and A.P. Ramesh, "Pipelined Architecture of 2D-DCT, Quantization and Zigzag Process for JPEG Image Compression using VHDL", *International Journal of VLSI Design and Communication Systems*, Vol. 2, No. 3, pp. 99-105, 2011.