FPGA Implementation of Convolution Neural Network

[1]Dr.A.Vino Vilmet Rose, [2] Mrs.N.Vini Antony Grace, [3]Mr.M.Aravindan, [4] Ms.L.Aquiline Lydia,.
[1]Assistant Professor, [2] Assistant Professor, [3] Assistant Professor, [4] Assistant Professor
[1]Electronics and Instrumentation Engineering,
[1]RMD Engineering College, Chennai, India

Abstract : ConvNets have several advantages as they operate with local receptive fields by performing convolutions: they share weights in the convolution .Secondly spatial subsampling/pooling is used to hierarchically reduce the input data size at each step of nonlinear computation. In this paper we present a FPGA implementation of convolutional neural networks to study the area consumption in FPGA

IndexTerms **- Artificial neural network, Convolution neural Network, Field Programmable Gate Array.**

I. INTRODUCTION

ANN introduction

The idea of Artificial Neural Networks (ANNs) was initially proposed by Warren McColluch and Walter Pitts [1] in 1943 when they come-up with "Threshold Logic Unit". ANNs are inspired by the working principle of brain where billions of neurons are networked together in a complex manner and are processing in parallel Implementing an electronic brain is one of the grand challenges [2].Several efforts are being made across the world to construct the first electronic brain capable of emulating a human-sized cortex [3][4]. ANN applications are growing very rapidly and to experiment with their full capability, large faster models comprising of huge number of neurons is required.

Applications such as computer vision, video processing in autonomous cars and medical devices which demand high accuracy and real time object recognition requires [5],[6],[7] need high computational requirements like convolution neural neural network . The high regularity of the structure, small size of convolution filters,easier to compute activation function makes the hardware implementation feasible.

II. CNN architecture design:

CNN involves the input size decreases from layer to layer and the number of filters increases. At the end of a network, a set of characteristics is formed that are fed to the classification layer (or layers), and the output neurons indicate the likelihood that the image belongs to a particular class [8],[9].

III.Implementation and Analysis

The FPGA implementation of CNN done with two FPGA families namely SPARTAN 3, device XC3S400 and Virtex 5 Device XC5VLX30 and the results have been tabulated.The Simulations were carried out using CIFAR -10 dataset consisting of of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

IV.Results:

	Device: 3s400tq144-4 Family SPARTAN-3
Number of Slices:	30 out of 3584 0%
Number of Slice Flip Flops:	53 out of 7168 0%
Number of 4 input LUTs:	38 out of 7168 0%
Number of IOs:	29

Number of bonded IOBs:	29 out of 97 29%
Number of GCLKs:	1 out of 8 12%
Minimum period:	5.664ns
Maximum Frequency:	176.554MHz

Table 1:FPGA implementation with SPARTAN-3

	Device: 5vlx30ff324-3 Family Virtex 5
Number of Slice Registers:	53 out of 19200 0%
Number of Slice LUTs:	48 out of 19200 0%
Number used as Logic:	48 out of 19200 0%
Number of LUT-Flop pairs:	59 out of 19200 0%
Number of IOs:	29
Number of bonded IOBs:	29 out of 220 13%
Number of BUFG/BUFGCTRLs:	1 out of 32 3%
Minimum period:	2.198ns
Maximum Frequency:	454.959MHz

Table :2 FPGA implementation with Virtex 5

As the device was changed from SPARTAN 3 to Vitex 5 the clock speeds have changed from 176.554Mhz to 454.959Mhz.The Virtex 5 implementation is better suitable for faster processing of images.

V. Conclusion:

The CNN implementation in FPGA is one of the needed requirements in todays world image processing . The FPGA implementation with two different devices was carried out and was found that Virtex gave better speed for realtime processing.

References:

[1] W.S. McCulloch, W. Pitts, ‘‘A logical calculus of the ideas immanent in nervous activity,’’ BullMath. Biophy. 5 (1943) 15–133.

[2] National Academy of Engineering(nae.edu), Grand Challenges for Engineering, www.engineeringchallenges.org, 2010

[3] D. Modha, et al, “Cognitive Computing: Unite neuroscience, supercomputing, and nanotechnology to discover, demonstrate, and deliver the brain’s core algorithms,” Communications of the ACM 54, 8, pp 62-71, August 2011.

[4] J. Schemmel et al, “A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neuron Modeling”, in Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, 2010.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521,no. 7553, p. 436, 2015.

[6] A. Shvets, A. Rakhlin, A. A. Kalinin, and V. Iglovikov, “Automatic instrument segmentation in robot-assisted surgery using deep learning,” arXiv preprint arXiv:1803.01207, 2018.

[7] A. Shvets, V. Iglovikov, A. Rakhlin, and A. A. Kalinin, “Angiodysplasia detection and localization using deep convolutional neural networks,” arXiv preprint arXiv:1804.08024, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[9] C. Szegedy et al., “Going deeper with convolutions,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 1–9.