International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

Probabilistic Principle Component Analysis based Feature Extraction of Embedded System Applications with Deep Neural Network based Implementation in FPGA

Prashant Bachanna[1], Baswaraj Gadgay[2] & Sayanti Chatterjee[3]
[1]Research Scholar, Department of ECE, Visvesvaraya Technological University (VTU), Regional Research Centre, Belagavi, Karnataka, India. e-mail: prashantece403@gmail.com
[1]Assistant Professor, Department of ECE, Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India. e-mail: prashantece403@gmail.com
[2]Regional Director, Visvesvaraya Technological University (VTU) Campus, Kalaburagi-585105, Karnataka, India. e-mail: bvgadgay@gmail.com
[3]Associate Professor, Department of EEE, Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India. e-mail: sayanti.chatterjee@iare.ac.in

Abstract— The study of hardware and software systems is of major are very important advent in new devices for communication and progress in system of security. In fast pace mobile and embedded devices application in every day's life leads some new emerging area for research in data mining field. In this we have some technologies which have demand and error free using the principle of component of PPCA. For Embedded system the applications of PCA is basically applied initially for the lessen the having different qualities especially being to simple of the data. PPCA which have the updated version of PCA which is surveyed by similarity measure. In this work, experiments are extensively carried out, using a FPGA based light weight cryptographic data set having benchmark set to check and illustrate the viability, competence, litheness which are reconfigurable embedded system which are having data mining . Which have FPGA are reconfigurable for the computing architectures for hardware and in neural network. FPGA using the multilayer Cascaded for neural network which are forward in nature (CFFNN) and Deep Neural Network also called as DNN with a huge neuron is still a thought-provoking task. This shortcoming leads to elect the FPGA capacity for a particular application we have used the method of implementation which has two neural network have been implemented and compared , namely, CFFNN and DNN. It can be shown that for reconfigurable embedded system, PPCA based data mining and Machine learning based realization can give more speed up less iteration and more space savings when we have compared it with the static conventional version.

Keywords- Data collection, Hardware-software systems, FPGA, Probabilistic Principle component analysis, Reconfigurable hardware.

I. INTRODUCTION

With the fast propagation of embedded and mobile system, an emerging area of research in data mining has unlocked into slender software code and small hardware architecture. In today's world data mining has become vital to every aspect including scientific research, medical diagnosis, marketing, biotechnology, multimedia, security, finance etc. Now-a-days, data handling and mining jobs are becoming computationally complicated and data intensive. These tasks need substantial data processing capacity. Additionally, in different situations, the real time data need to be handled to earn the actual aids. These restraints have a huge impression on the performance accuracy and speed in the embedded system applications. To mollify the necessities and constrictions in the portable, embedded devices, and also to speed up the process of the application on these devices its domineering for the integrate of some hardware into software and hardware system designs. The designed hardware which are specifically designed the delivers higher performance speed, area efficiency and less power consumption, [1,2] that are compared to the corresponding software on microprocessors.

In the recent years, machine learning has become leading field in innumerable research applications, containing human activity recognition. Nevertheless, with the cumulative accuracy desires and complication for practical solicitations, device speed and the cost of power make the implementation impossible. To overcome of those drawbacks, expand the efficiency as well as the low power consumption have been provided to reconfigurable use of hardware has produced attentiveness. Proposed Design, Implementation and validation of an architecture are developed to overcome the deficiency and enhanced the speed and performance of the proposed design as well as sustaining very low power consumption. This proposed

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

work objectives are to develop and estimate a comprehensive reconfigurable that supports the run-time through machine learning (ML) approaches. The key awareness of this architecture was based on ML algorithm platform that is enhancing classifiers that running on a general-purpose processor, by visibly mapping using computationally intensive parts that execute the re-configurable hardware. Through this approach, not only enhance the architecture performance but also it will increase the embedded application that will build up the units which will acting as co-processer of general-purpose systems. Machine learning it is a specialized field that involves teaching machines how to learn with the ability to rapidly process large amounts of data through mathematical calculations, machine learning has emerged as a powerful tool in recent years. However, a new generation of computing approaches known as reconfigurable computing is now poised to take machine learning to even greater heights. The computing offers the speed of application of IC flexibility of processors. Among the most popular execution platforms for reconfigurable computing are reconfigurable instruction set processors, which have been developed through a variety of design alternatives. The effectiveness of these processors is largely dependent on the design of their configuration units, which are dynamically control the updating of the processor's configurations. To enhance the performance of reconfigurable processors, has the configuration design that is developed that maximizes the reusability of existing configuration streams. This design enables the loading of the most optimal configurations, resulting in greater efficiency and improved performance. In some basic application fixed design of processor will be a good choice. However, the unknown applications have the diversity of algorithms that the case of it where having fixed standard will fail to deliver the required process of speed. It is very difficult task to create a specialized hardware. Machine Learning Application with the diversity algorithm requires much specialized hardware are more expensive than the reconfigurable solutions. Reconfigurable computing is very important for research of computing paradigms. Reconfigurable computing has performance, flexibility of the gaining on a single computing system. The performance of the systems is crucially dependent on the configuration which are caused by the management unit. The configuration management that speeds up the computational power of reconfigurable computing. The control and management techniques which help to improve the technology [1]. Configuration are used to reduce the emerging concepts of multiple switching, partial reconfiguration, configuration cloning and configuration pipelining. This are the dimensions of power in the reconfigurable computing system. In paper [2], the Reconfigurable computing is comprised of three primary aspects: architectures, design methods, and applications. In terms of architectures, current trends are focused on coarse-grained fabrics, heterogeneous functions, and soft cores. Coarse-grained fabrics utilize larger blocks of reconfigurable logic, while heterogeneous functions allow for greater customization and optimization of specific tasks. Soft cores, on the other hand, enable the use of pre-existing processor architectures that can be customized to meet the specific needs of a given application applications like media processing and numerical processing and embedded applications. In paper [3] Their investigation revolves around exploring the impact of FPGA embedded array architectures on their ability to execute logic. Their primary focus is on architectures that comprise multiple sizes of memory arrays, and their findings indicate that such heterogeneous architectures have the result in much deep for the implementations of logic compared to with single size memory array. Zippy it is a hybrid CPU which utilizes multi-context reconfigurable array and stands out with its built-in hardware virtualization support. In terms of hardware virtualization, Zippy employs a specific approach known as virtualized execution., in paper [4] we focused on an approach which is termed to temporal partitioning. In this paper [5-7] the digital electronics part of technologies that offer enormous scope of the computing the power and has been a restoration with in field of AI there is a specific focus on deep learning (DL). In the existing literature [10-13], high performance calculation methods for FPGA have been discussed which helps the current researcher to find the proper selection pf algorithm. Software simulation of a self-organizing learning array system which is the backbone of neural network has been illustrated by many researchers and has been elaborated in the papers [14-16]. The primary goal in this work is to provide optimized solution for embedded system to boost computational speed and data rigorous applications for example application of data collection on mobile or embedded devices. In this novel optimized method for data extraction has been introduced for architectures of embedded hardware using the Probabilistic principal component analysis (PPCA). PPCA basically a statistically data handling technique to decrease of data. In this method the true data has been transformed in new dataset by unaltering the key features. In the literature PCA [25,27,28] has been already exist. PPCA has been for better result. This PCA and PPCA based data mining has been incorporated in image processing of biometric data to augment the performance and to minimize memory access of embedded platforms. The second part of the work deals with the data recognition and thereby classification for biometric images. This is the extreme efficient benchmark which can be elucidated using NN [8] [9]. In the work [20-22], different neural network-based implementation has been advocated for different systems. In this paper the Cascaded feed forward NN and deep NN has been used to study as a model for image processing. The accurate recognition of offered pictorial objects should be robust under scaling, translation and rotation of the input [22]. To fit in these

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

attributes into a neural network, the choice of the algorithm should be proper. The second section of this paper deals with features, advantages of FPGA based reconfiguration. The section III discussed about the PCA and PPCA based data Mining followed by the CFFNN and DNN algorithm in the fourth unit. The fifth unit is about the results and wrapped up with conclusion.

II. FPGA-BASED METHODS FOR RECONFIGURATION: A BRIEF IDEA

It deals with the data analyzing and discussion, features, pros, and cons of FPGA reconfiguration methods. The important thing is the context of reconfiguration methods has been revisited here.

- **Configuration bitstream**: It can be defined as a binary file which contains the programmable binary locations all of the FPGA to organize the routing resources. At nascent stage, the bitstreams are deposited peripheral non-volatile memory. In the second stage it is downloaded from memory to the chip and applied hardware circuitry of the chip to configure it.
- **Inactive context:** The next step is inactive context when inactive data are held in reserve onto chip for future. The data are utilized repeatedly to reconfigure the chip as needed.
- **Active in context**: Unlike the inactive context, in this case all the bit files after download of configuration are utilized to the reconfigure the chip immediately. This phenomenon is known as active in context and does not need mean it is running. With the increment in dimensions, the distance becomes less accurately mostly for spatial data, this idea is always true. The process of clustering is usually employed to group objects together based on the similarity of their attribute values. However, if there are numerous attributes involved, there is a high probability that some of these attributes or features may be redundant, thereby impacting the calculation of proximity measures and ultimately the clustering results [4, 24]. This can be mitigated by identifying correlations among subsets of features.

III. PCA AND PPCA BASED DATA MINING

Data collection which is an important area for the research as as they are many domains in different fields and that can make use of it to through large volumes of data to discover useful patterns and valuable knowledge. Many of the data collection algorithms are necessary for the complex computations and speed is a major concern. New things such as cloud and grid to leverage the power of parallel and distributed computing. However, they exist many applications, such as hand held, portable, hardware and software devices, that are more accessible and suitable for to be processed in traditional computing environment. An alternative is to take advantage of the recently available is reconfigurable FPGA hardware. Our work tells us that it investigates us the use of using such an approach for compute intensive data mining operations, specifically, the often-used PCA. The performance gain or speedup resulting from the use of hardware over software is computed using the following definition.

$$Speedup = \frac{Baseline\ Execution\ Time\ (Software)}{Improved\ Execution\ Time\ (Hardware)} \quad (1)$$

PCA are defined as a tool for preprocess of the data. Here the version of PCA introduced in the search of the most applicable combination of variables to minimize the data redundancy. Here PPCA is used in the casted to shrink the bulky data set, consequential from the test. PPCA simultaneously can optimize the information loss. Another important aspect of this method is reduction of dimensionality by clustering the small data sets keeping the information unaltered. The main motivation behind this is that smaller data sets are easier to explore visualize and analyze, which in turn can be used in neural network and other machine learning algorithms without superfluous variables increasing the processor speed with less memory. It is also applied when missing data or for multidimensional scaling are required. The algorithm can be jotted down as follow:

Step 1: The continuous initial variables range standardization.
Step 2: The covariance matrix computation to recognize correlations
Step 3: The eigenvectors and eigenvalues derivation from the covariance matrix
Step 4: The principal components identification
Step 5: A feature vector creation to adopt suitable principal components and discard others.
Step 6: The data along the principal components axes alteration
The detailed algorithm used in this work is furnished below.

Input: *v:* Raw data,
e: Error rate
Step 1: Computation the empirical mean shape

$$\bar{v} = \frac{1}{k}\sum_{j=1}^{k} v_j \quad (2)$$

Step 2: Computation of the co variance matrix *C*, where

$$C = \frac{1}{K}\left[(v_1 - \bar{v})(v_2 - \bar{v})..\left(v_j - \bar{v}\right)..(v_k - \bar{v})\right]\left[(v_1 - \bar{v})(v_2 - \bar{v})..\left(v_j - \bar{v}\right)..(v_k - \bar{v})\right]' \quad (3)$$

Step 3: Decomposition of the Co variance of the Matrix

$$C=UVD^T \quad (4)$$

Step 4: Calculation of the Eigen values of *D* matrix and rearranging which is in the descending order where i[th] Eigen value:

$$\lambda_i = d_{i^2} \quad (5)$$

Step 5: for m=1: (d-1)

$$\text{Error E}= \frac{1}{k}\sum\left\|v_j - U_j U_{j^T}\left(v_j - \bar{v}\right) - \bar{v}.\right\|^2 \quad (6)$$

Step 6: Compute **S**= $\frac{\frac{1}{k}*1}{d-k\sum_{i=k+1}^{d}\lambda_i}$ (7)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

$$\text{Ufinal} = U_k\sqrt{(\lambda_k - S * I)} \quad (8)$$

IV. Cascaded Feed Forward Neural Network and ML Algorithm

CFFNN is basically a Feed Forward neural network. Figure 1 illustrates a 5-layer neural network. In this paper, Feed Forward networks have been cascaded with more than 2 layers of network. The CFNN are used to utilized for any kind of contribution. The advantage of this method can be summarized that it can be deployed among entry and exit level.

The relationship between the input and the output is direct. We can use this model which is in equation 9.

$$y = \sum_{i=1}^{n} f^i w_i^i x^i + f^o \quad (9)$$

f0 : the activation function on the output layer

f_j^h : an activation function on the hidden layer

w_j^0: the weight of the jth neuron at the output layer

w_{ji}^h : the weight of the jth neuron at the hidden layer

w_i^i :the weight of the ith neuron at the input layer

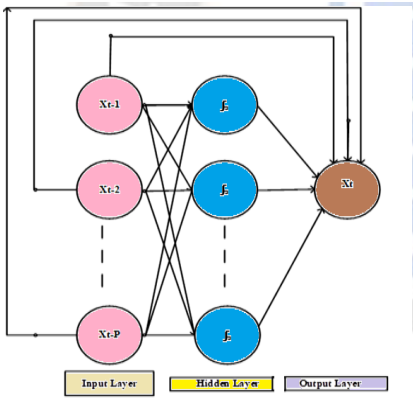

Figure 1. CFFNN architecture-5 layers.

Equation 9 can be rewritten as equation 10 in presence of an additional bias the contribution layer.

$$y = \sum_{i=1}^{n} f^i w_i^i x_i + f^o\left(w^b + \sum_{j=1}^{k} w_j^o f_j^h\left(\sum_{i=1}^{n} w_{ji}^h x_i\right)\right) \quad (10)$$

ω^b for the weight of bias from its input.

Figure 1 discussed about the architecture details for CFFNN algorithm elaborated down.

A. Selecting Algorithm of CFFNN Technique

Step 1: Initialize k and starting position

Step 2: Determine the initial weight gradient g and α_k

$$g^{(0)} = \frac{de}{dw^0} = \frac{de}{dw_{|w=w^0}}$$

$$= \left|\frac{de}{dw_1^{(0)}} \cdots\cdots \frac{de}{dw_s^{(0)}}\right|^T \quad (11)$$

$$\alpha_k = argmin_{\alpha \geq 0} e\left(w^{(k)} + \alpha d^k\right)$$

$$= \frac{-g^{(k)T} d^{(k)}}{d^{(k)T} Q_{d(k)}} \quad (12)$$

Step 3: Fix $\Omega^{(k+1)}$ uses equation 6:

$$\Omega^{(k+1)} = \Omega^k + \alpha_k d^{(k)}$$

Step 4: Optimize $g^{(k+1)}$

$$g^{(k+1)} = \frac{\partial e}{\partial w^{(k+1)}} \quad (13)$$

if $g^{(k+1)} = 0$, the optimal weight is w(k+1)

>> end of the loop.

otherwise

Step 5: Determine $\beta_k \wedge d^{(k+1)}$ using equation 7 and 8:

$$\beta_k = \frac{g^{(k+1)} Q d^k}{d^{(k)T} Q d^k} \quad (14)$$

$$d^{(k+1)} = -g^{(k+1)} + \alpha_k d^{(k)} \quad (15)$$

Step 6: k = k+1: back to stage 3

The proposed multi-layered cascade NN model is a algorithm to trace the maximum power points for the hardware and software system as shown in fig 2.

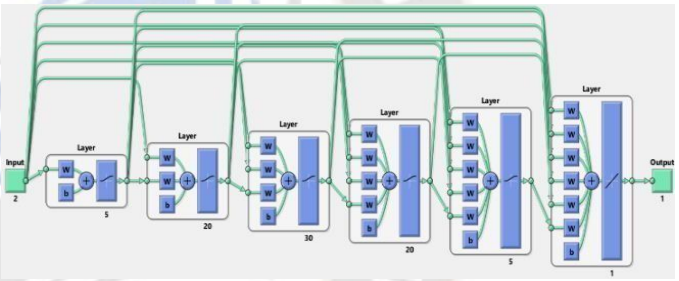

Figure 2. CFFNN architecture-multi layers.

V. Using Deep Neural Network(DNN)

Neural Network is the one of the newest technologies in machine learning era. This is a MISO system where as individual neuron is a mapped with multi-inputs but one output is inversely with input. Figure 3 describes the DNN architecture is expressed in the form of mathematical which is shown below and modified in equation 16.

$$y = f^o\left(\sum_{j=1}^{k} w_j^o f_j^h\left(\sum_{i=1}^{n} w_{ji}^h x_i\right)\right) \quad (16)$$

Where f^0 is the layer of production

f_j^h : Hidden layer initiation function,

$$y = f^0$$

The production weight is $w^b$and bias layer is unseen is w_j^b.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

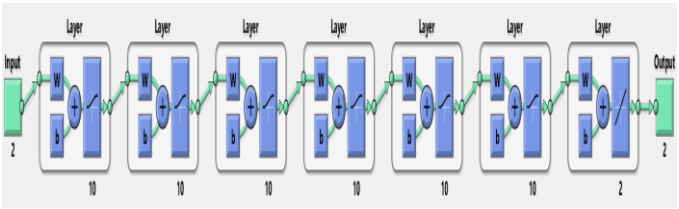

Figure 3. DNN architecture.

VI. RESULTS

The calculation of different stages of PPCA are namely (i)Frist stage mean stage, (ii) second stage is Covariance Matrix and Eigenvalue Matrix which is the third stage, and PC Matrix: stage 4 has been performed experimentally as discrete entities, with changeable vector number and data sizes. The Monte-Carlo run of 100 execution time has been performed for each stage and the average execution time is offered. Table 2 demonstrates different execution times and all four stages of PCA, PPCA has been deliberated and compared here. Figure 3 discussed the comparison of (a) covariance matrix (b) Eigen value matrix and (c) total execution clock cycle with respect to data size.

TABLE I. TABLE TYPE STYLES

Data Size	No. of vector	Execution time in AXI(in million)									
		Stage 1		Stage 2		Stage 3		Stage 4		Total	
		PCA	PPCA	PCA	PPCA	PCA	PPCA	PCA	PPCA	PCA	PPCA
30528 84	477	.9365	.9355	6.15	6.07	1.803 / 332	1.803/33 1	1.203	1.118	1.8230	1.7843
61 12	956	1.875	1.795	12.31	12.19	1.271 / 235	1.270/23 5	2.50	2.32	14.325	15.084
	1433	2.807	2.738	18.45	18.23	1.402 / 258	1.399 / 257	3.61	3.58	21.192	24.327

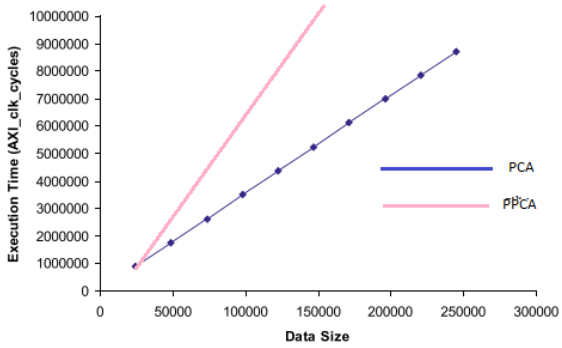

Figure 4. Matrix covariance and comparison for execution time vs data size for DRH

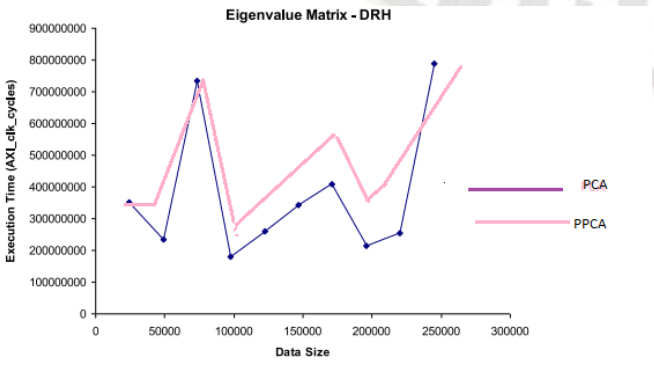

Figure 5. Eigen value matrix comparison for execution and data size for dynamic

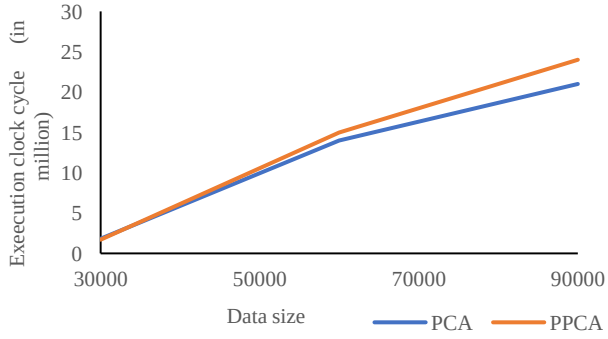

Figure 6. Comparison of total execution clock cycle w.r.t data size

Figure 4,5 and 6 compares results of PCA and PPCA based data mining with respect to clock cycles, covariance matrix and eigen value matrix. In this work, dynamic reconfigurable hardware architecture has been introduced for PPCA. The partial method of reconfiguration can be deduced from the following.

mean → covariance matrix → eigenvalue matrix → PC matrix computations

Design helped to save area significantly (upto 79%) subsequently the same chip area is being reused (using reconfiguration of the hardware) in all the four stages of calculation decisive for embedded devices with limited hardware.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

On a standard Xilinx system, the kernels have been run directly from the xclbin file. It can be seen that the time used for feedforward layer for CFFNN and DNN is negligible when it is compared with the time consumed to process the for NN layers with respect to the speed up of process. The image processing of the biometric data has been taken into account to prove the efficacy of these methods. Table 2 gives the details of the comparison.

TABLE II. TABLE TYPE STYLES

Attributes	CFFNN	DNN
Segmentation clks for one large receptive field (St)	2497 clks	2520 clks
Calculation clks for one large receptive field (Ct)	682 clks	703 clks
Feed forwared layer clks for each image (Ft)	294 clks	300 clks
One image processing clks	318494 clks	32434 clks

clk : clock period time(1/50MHz=20 ns in FPGA - XC3S200)
N: Number of large receptive fields in one image.

VII. CONCLUSION

- In this work, PPCA algorithm has been used to familiarized dynamic solution of embedded hardware solutions for probabilistic principal component analysis designed. PPCA has been elected to apply in the first stage of this research. The results are compared with PCA based theorem which is available in literature.
- In the initial stage of work, an investigation on dynamic embedded platforms has been platform especially on image processing-based FPGAs. Then the performance of PCA and PPCA has been tested and compared by different stages of dynamical run like mean of covariance, eigen value decomposition, probabilistic principal components computations and total clock used respectively. T
- The total time of execution has been calculated by adding the abovementioned times of four stages. It can be concluded that the selected embedded architecture has been grasped to a perceptible speedup. It can be also shown that PPCA gives more speed up than PCA. Furthermore, the reconfigurable embedded system design has about 92% area saving compared to old design which is based on old and static hardware.
- A novel CFFNN and deep learning algorithm has been successfully implemented in this work for unstructured data for a comprehensive reconfigurable FPGA-based architecture that supports the run-time and the dynamic distribution of multiple accelerators.

With the propagation of embedded devices, portable and/ or mobile technology, data mining-based applications have initiated their mode in these strategies. More likely they are kinds of computationally complex and data privacy which are extremely intensive applications needs to construe vast amount of information with multifaceted algorithms.

The future scope of this work:

1. It is planned to use support vector machine algorithm for feature extraction from the analysed data.
2. Machine learning technique should be used for the fault detection in the data mining. If any fault is present, it can be detected quickly

REFERENCES

[1] Aqeel Iqbal, M., Farooque Azam, Uzma Saeed Awan ,Performance Enhancement Techniques for Modern Reconfigurable Computing Systems, International Journal of Computer Applications (IJCA), 27(09): 33-38.

[2] Todman, T.J., G.A. Constantinides, S.J.E. Wilton,Reconfigurable Computing: Architectures, Design Methods, and Applications. IEE Computers and Digital Techniques, 152(02): 193-207.

[3] Steven I.E. Wilton, Implementing Logic in FPGA Memory Arrays: Heterogeneous Memory Architectures, (FCCM' 02), pp: 142-147, Napa Valley, Calif, USA.

[4] Plessl, C. and M. Platzner, 2005. Zippy - A coarse-grained reconfigurable array with support for hardware Virtualization, (ICASAP' 05), pp: 213-218.

[5] Francisco Barat, R. and Lauwereins G. Deconinck, Reconfigurable Instruction Set Processors from a Hardware/Software Perspective, IEEE Transactions, on Software Engineering, 28(9): 847-862.

[6] Reiner Hartenstein, A Decade of Reconfigurable Computing: a Visionary Retrospective, (DATEC' 01), pp: 642-649, Munich, Germany, 2001.

[7] Y. Bengio et al., "Learning deep architectures for ai," Foundations and trends® in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009. 1

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015. 1

[9] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning. MIT Press Cambridge, 2016, vol. 1. 1

[10] M. C. Herbordt, T. V. Court, Y. Gu, B. Sukhwani, A. Conti, J. Model, and D. DiSabello, "Achieving high performance with fpga-based computing, " Computer, vol. 40, pp. 50-57, 2007.

[11] J. L. Rice, K. H. Abed, and G. R. Morris, "Design heuristics for mapping floating-point scientific computational kernels onto high performance reconfigurable computers, " Journal of Computers, vol. 4, pp. 542-553, 2009.

[12] T. El-Ghazawi, D. Bennett, D. Poznanovic, A. Cantle, K. Underwood, R. Pennington, D. Buell, A. George, and V. Kindratenko, "Is high-performance reconfigurable computing the next supercomputing paradigm?" in Proceedings of the 2006 ACM/IEEE conference on Supercomputing, 2006.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6771
***Article Received**: 20 March 2023 **Revised**: 30 April 2023 **Accepted**: 21 May 2023*

[13] R. e. a. Baxter, "High-performance reconfigurable computing in Proceedings of the second NASA/ESA Conference on Adaptive Hardware and Systems, 2007.

[14] J. A. Starzyk and Z. Zhu, "Software simulation of a self-organizing learning array system". The 6th IASTED Int. Conf. Artificial Intelligence & Soft Comp (ASC 2002).Canada.

[15] J. A. Starzyk and T-H. Liu, "Design of selforganizing learning array," IEEE Int. Symp. on Circuits and Systems (ISCAS), Bangkok, Thailand, May 2003.

[16] H. Ritter and G. Heidemann, "A neural 3_D object recognition architecture using optimized Gabor filters", Proceeding of 13th International conference on pattern recognition (1996), IEEE computer society press, IV, pp.70-74.

[17] Basil. Mahmood and Shefa. Dawwd," High Performance Image Recognition System Based Neural Network", 1st International Conference on Telecomputing and Information Technology, ICTIT 2004, Amman-Jordan, pp. 23-28.

[18] R. Molz, P. Engel, F. Moraes, L. Torres, M. Robert, "Codesign of fully parallel neural network for a classification problem", SCI'00 : 5th World Multi-Conference on Systemics , 01/01/2000, pp. 601-605,Cybernetics and Informatics,Pages: 133 - 150 ,Year of Publication: 2003.

[19] R. Molz, P. Engel, " A New proposal for Implementation of Competition Neural Networks in Analog Hardware"", Vth Brazilian Symposium on Neural Networks, Belo Horizonte-MG, pp. 186-

[20] Attila Hidvegi, "Implementation of Neural Networksin FPGAs", Available: http://www.sysf.physto.se/~attila/ANN.pdf

[21] Rob Chapman, Steven Sutankayo, "Implementing Artificial Neural Network Design", Final Report April 7, 1998, http://citeseer.ist.psu.edu/487920.html

[22] Li, X., Moussa, M., Areibi, S., "Arithmetic formats for implementing Artificial Neural Networks on FPGAs" Canadian Journal on Electrical and Computer Engineering (in print) (2005).

[23] K. Korekado, T. Morie, O. Nomura, "A convolutional Neural Network VLSI for image Recognition Using Merged/Mixed Analoge-Digital Architecture", Journal Title: Knowledge-Based Intelligent Information & Engineering Systems,Date: 2003.

[24] Rob Chapman, "Using A Neuroprocessor Model for Describing Artificial Neural Nets", Project Report for EE563, Neuroprocessor Modeling, December 16, 1997.

[25] Shahrouzi, S. Navid, and Darshika G. Perera. "Dynamic partial reconfigurable hardware architecture for principal component analysis on mobile and embedded devices." EURASIP Journal on Embedded Systems 2017.1 (2017): 1-18.

[26] Chatterjee, S. (2022). Fault detection for a nonlinear switched continuous time delayed system using machine learning and self-switched UKF. Journal Européen des Systèmes Automatisés, Vol. 55, No. 2, pp. 245-251. https://doi.org/10.18280/jesa.550212

[27] D. G. Perera. "FPGA-Based Reconfigurable Hardware for Compute Intensive Data Mining Applications", 2011 International Conference on P2P Parallel Grid Cloud and Internet Computing, 10/2011

[28] Shahrouzi, Seyed Navid. "Optimized Embedded and Reconfigurable Hardware Architectures and Techniques for Data Mining Applications on Mobile Devices.", University of Colorado Colorado Springs, 2018.