FPGA IMPLEMENTATION OF DEEP LEARNING APPROACH FOR EFFICIENT HUMAN GAIT ACTION RECOGNITION SYSTEM

[1]P.N.Palanisamy, [2]N.Malmurugan, [3]J.Sampathkumar

[1]Assistant Professor,Dept of Electronics and Communication Engg, Mahendra College of Engineering, Salem, India.E-mail: pnpalanisamyece@gmail.com
[2]Principal, Mahendra College of Engineering, Salem, India
E-mail: n.malmurugan@gmail.com
[3]Assistant Professor, Dept of Electronics and Communication Engg, Mahendra College of Engineering, Salem, India.E-mail: sampathkumar1980@gmail.com

Abstract- Human gait analysis is an appreciable method for activity recognition and monitoring systems depending on their movement pattern. It has a big part to play in monitoring systems, clinical and security uses. Even so, the existing gait method could have varying distinctions owing to the shifting positions and ambiguity characteristics obtained during the gesture recognition. Implementation of relevant prediction performance remains a persistent concern for researchers because of the different characteristics of human gaits. Hence to solve the problem, hybrid classifier of Convolutional Neural Networks Integrated with Long Short Term Memory has been proposed. To reduce the computational overhead; the paper proposes the hardware-software co design methodology for implementing the hybrid learning models for gait video processing. The experimentation is conducted using ARM CPU with ARTIX-7 FPGA with calculated and analyzed. The implementation is carried out using OPENCL programming in which the proposed methodology finds more suitability in implementing the hybrid deep learning models with less area and power consumption than the existing baseline architectures.

Keywords- Human Gait Video Analysis, Long Short Term Memory, Convolutional Neural Networks, FPGA, ARM CPU, OPENCL, baseline

1 INTRODUCTION

Frameworks for person identification are partitioned into two classes: prototype-free and prototype based approaches [1]. Prototype-free methods use a group of static or dynamic composite membranes by observing parts of the body, including limbs, legs, sides, and calves. Prototype methods were regarded as irreducible and scale-independent. Such benefits are important for efficient reasons since it is not necessary for comparison and check sequences to be collected via an equal perspective. Prototype methods, indeed, have significant drawbacks. These are prone to the consistency of the gait databases, and therefore elevated gait images captured are needed in order to attain maximum precision. The additional drawback of the prototype method is its detailed analysis and high time costs due to the experiment to determine variables. Prototype- free techniques focus on the contour lines of the silhouettes or even the whole orientation of the human parts[2-4], instead of the whole moving image or some portion of the anatomy. Prototype -free methods do not take into account the consistency of silhouettes and also have the benefits of reduced computing costs relative to prototype techniques. With an advent of deep learning algorithms, gait recognition has reached its greater heights[10-12]. For an efficient implementation, the paper proposes the Convolutional Neural Networks integrated with the Long Short Term Memory along with its FPGA Implementation. The contribution of the paper is as follows

1. The powerful d CNN has been implemented for an effective gait feature extraction. This feature forms the unique set of datasets, which overcomes the intra and inter variations.
2. Second the adoption of boosted LSTM algorithm for an effective classifier. LSTM is incorporated for the better classification and tracking in which it can be suitable for scalable datasets.
3. Finally, the FPGA Implementation of the above mentioned algorithms for an effective recognition.

The article is framed as follows as: Section-2 covers the related works of the different authors about the FPGA based CNN. The working mechanism of CNN and LSTM has discussed in the Section -3. The conventional methodology of implementing the CNN _LSTM in FPGA is presented in Section-4 and results are explained in Section-5. Lastly the article is concluded in section -6.

2 RELATED WORKS

C. Farabet(2010) presented a scalable hardware architecture to implement large-scale convolutional neural networks and state-of-the-art multi-layered artificial vision systems. This system is fully digital and is a modular vision engine with the goal of performing real-time detection, recognition and segmentation of mega-pixel images. We presented a performance comparison between a software, FPGA and ASIC implementation that shows a speed up in custom hardware implementations[8].

S. Chakradhar (2010) drived the design of a new architecture for CNN. First, CNN workloads exhibit a widely varying mix of three types of parallelism: parallelism within a convolution operation, intra-output parallelism where multiple input sources (features) are combined to create a single output, and inter-output parallelism where multiple, independent outputs (features) are computed simultaneously. Workloads differ significantly across different CNN applications, and across different layers of a CNN. Second, the number of processing elements in an architecture continues to scale (as per Moore's law) much faster than off-chip memory bandwidth (or pin-count) of chips. Based on these two observations, we shown that for a given number of processing elements and off-chip memory bandwidth, a new CNN hardware architecture that dynamically configures the hardware on-the-fly to match the specific mix of parallelism in a given workload gives the best throughput performance. CNN compiler automatically translates high abstraction network specification into a parallel microprogram (a sequence of low-level VLIW instructions) that is mapped, scheduled and executed by the coprocessor. Compared to a 2.3 GHz quad-core, dual socket Intel Xeon, 1.35 GHz C870 GPU, and a 200 MHz FPGA implementation, our 120 MHz dynamically configurable architecture is 4x to 8x faster. This is the first CNN architecture to achieve real-time video stream processing (25 to 30 frames per second) on a wide range of object detection and recognition tasks[9].

Xiaofan Zhang et al. [6] (2017) have introduced a design architecture for DNNs which use customizable IP addresses in the neural network layer, including a new model audit tool for allocation of resources management (REALM). Perhaps it performs advanced cognitive architecture and fixed-point print retraining that will further enhance FPGA resolution. This illustrates the architectural structure of an enticing long-term video input circulation network. The Xilinx VC709 on-board drive uses 17.5x less power per image, 3.1x faster than the NVIDIA K80 and 4.75x faster than the Intel Xeon.They also implemented a REALM resource allocation plan that provides conceptual guidelines for all levels of allocation of resources during the smallest general unit era. The proposed HLS IP instance for the implementation of the LRCN used the resource allocation guidelines to achieve a design that exceeds both the GPU and CPU workloads in terms of power and latency.

Kamer Kayaer et al.[7] (2008) have introduces a unique processor architecture that implements a discrete neural cellular network (DT-CNN) to be implemented in FPGAs. This architecture is designed to manipulate real-time video images using 3x3 CNN models using only internal memory. The lack of external memory lowers the system's expense and complexity. This model is focused on a completely carrier cell used to model a large number of neurons on CNN.The data input needs to send data to the FPGA in an innovative way, such as VGA, DVI, advanced camera or digital interface CMOS camera. Interconnected video signals such as PAL, SECAM and NTSC are not acceptable for such frameworks. Video representations are interpreted and displayed at the same time as input intensity values. The output of the system is aligned with the input with some video delay along with the delay due to the system pipeline model. The new structure is also implemented in the Xilinx Virtex-II FPGA on the Celoxica RC203 card. The number of Euler iterations needed for the computation of the actual output shall be calculated based on the number of processing units (PUs).

3 PRELIMINARY WORKS

In this paper, we deployed FPGA Based CNN algorithm with LSTM for human activity anamoly detection. Generally, CNN layer comprises of input, pooling and fully connected layers. The input layer includes the gait features denoted as ‘W’ in I^(k*d*m) were k,d, m are the length, width and height of the input layer. The activation function ‘S’ is represented by I^(l*l*m*n) were ‘l’ is the size of the convolution kernel and ‘n’ is the number of the ouptut scores. We considered the proposed CNN includes stride in range of one’s. The ouput feature maps are represented as equation below.

3.1 CNN Learning Process

The parameters of Scattered Convolutional Neural Networks (SCNNs) are learned as initial decomposition and fine-tuning stages.

$$P(y,x,j) = \sum_{i=1}^{m}\sum_{u,v=1}^{S} I(i,j,k)S(y+u-1, x+v-1, i) \quad (1)$$

The above equation (1) represents output feature maps obtained after convoluting the input images with the filters and activation layer. The main objective of Scattered CNN is to reduce the complexity involved in the traditional CNN by its scattered version which is based on multiplication of scattered matrices (tensors). For this purpose, we first transform the convolutional networks and decomposed the every matrices P into Q, R and S whose mathematical equations are given below

$$Q(i,j,k,n) = \sum_{k=1}^{m} I(i,j,k,n)\ P(k,j) \ (2)$$

$$R(y,x,i) = \sum_{k=1}^{m} P(i,k)I(y,x,k) \quad (3)$$

$$S(u,v,i,j) = \sum_{k=1}^{wi} I(k,j)wi(u,v,k) \ (4)$$

After decomposition, the equation (1) is modified which is then given below

$$T(y,x,k) = \sum_{u,v=1}^{w} wi(u,v,k)J(y+u-1, x+v-1, i) \ (5)$$

The factorization in both formula (3) and formula (5) can be treated as matrix factorization problem if we transform the convolution kernel from tensor to matrix along the decomposition dimension. The convolutional neural network are used to extract the silhouette images along with the different features.

3.2 Long Short Term Memory

This network is composed of cell, input gate, output gate, forget gate. Cell Units are considered LSTM memory to recall qualities throughout times. Let xt, the performance of the hidden layer is ht and its previous output is ht−1, the input state of the cell unit is Ct, the output impedance of the cell is Gt and its prior state is Gt-1, the output of the forget gate is j t. The LSTM cell arrangement suggests that these Gt and Ht are transferred towards the next neural network throughout the RNN.LSTM compares the information from the previous machine with the new input state where the output and forget gates are being used to refresh the information. We just use equivalent circuit in calculating Gt and Ht. Next, measure the three gate states as well as the cell input status, the input gate:

The input gate is specified as

$$j_t = \theta(G_l^i.O_t + G_h^i.e_{t-1} + s_i) \quad (6)$$

The forget gate is expressed as

$$T_f = \theta(G_l^f.O_t + G_h^f.e_{t-1} + s_f) \quad (7)$$

Output gate performance is estimated as

$$T_o = \theta\,(G_l^0.O_t + G_h^o.e_{t-1} + s_o) \quad (8)$$

Entry cell value is given by

$$\widetilde{T_C} = \tanh(G_l^C.O_t + G_h^C.e_{t-1} + s_C) \quad (9)$$

Where $G_l^0, G_l^f, G_l^i, G_l^c$ are the weight matrices attaching the entry gates to output nodes, while $G_h^i, G_h^f, G_h^o, G_h^C$ are indeed the weight matrices attaching the entry gates to the hidden units. Even s_i, s_f, s_o, s_C are bias variables and tanh is called hyperbolic function. The second cell output condition is determined and described as follows

$$T_C = k_t * \widetilde{T_C} + T_f * T_{t-1} \quad (10)$$

Likewise hidden layer output is estimated as

$$e_t = T_o * \tanh(T_C) \quad (11)$$

4 FPGA IMPLEMENTATIONOF PROPOSED ALGORITHM

For an effective implementation of the proposed CNN -LSTM architecture in FPGA, the proposed system incorporates the hardware- software strategy in which the CNN‘s convolutional layers are implemented in the programmable FPGA and other layered architectures are moved in to the software design.

Figure 1 shows the proposed architecture which incorporates the hardware -software mechanism. The CPU converts the videos to frames and frames are sent to the FPGA for the further processing. The ARM CPU is incorporated and it is interfaced with the FPGA using the AXI -Connect Buses. The general matrix multiplier units (GMMU) are used for multiplying the input frame vectors and filter coefficients. The GMMU uses the eight pipelined floating point multipliers which are connected with the different block RAMS(BRAMS) which contains the input feature matrix and output feature matrix. The collected feature maps are accumulated using the pipelined adder trees and stored in the BRAM. Then the output interfaced with the CPU which again runs the LSTM networks for better classification and recognition of human activity. The CPU takes responsibility for feeding the correct addresses to the input and output matrix for an exact processing .Once the signal is received from the CPU, GMMU starts its operations and stores it in the memory for further processing.

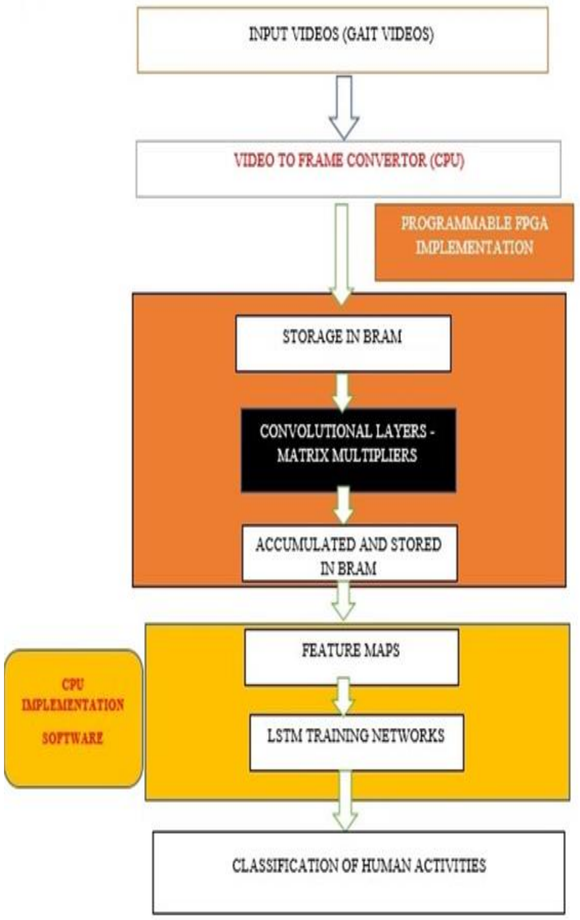
INPUT VIDEOS (GAIT VIDEOS)
VIDEO TO FRAME CONVERTOR (CPU)
PROGRAMMABLE FPGA IMPLEMENTATION
STORAGE IN BRAM
CONVOLUTIONAL LAYERS - MATRIX MULTIPLIERS
ACCUMULATED AND STORED IN BRAM
FEATURE MAPS
CPU IMPLEMENTATION SOFTWARE
LSTM TRAINING NETWORKS
CLASSIFICATION OF HUMAN ACTIVITIES

Figure 1 Hardware-Software Strategy for the Proposed CNN-LSTM architectures for Input Video Sequences.

5 RESULTS AND DISCUSSSION

5.1 Performance Metrics

To evaluate and validate the proposed intrusion detection system using the Proposed architecture, we have calculated the different parameters such as Accuracy, Sensitivity Selectivity, Specificity and evaluate the performance with the model.CASIA datasets were used for the validation and testing.For each datasets, 70% has been taken as the training data and 30% as the testing data. The performance metrics has been estimated by the following mathematical expression

Accuracy=DR/TNI x100

Sensitivity=TP/(TP+TN)

Specificity=TN/(TP+TN)x100

Where TP and TN denotesTrue Positive and True Negative values and DR & TNI denotes the Number of Detected Results and the total iteration count.

Figure 2 shows the accuracy of the proposed algorithm. It is found that accuracy is 96.5 % for recognizing the human activities. The resultant accuracy, specificity and sensitivity have been calculated based on the confusion matrix which is given in Figure 3. Figure 4 and 5 shows the specificity and sensitivity analysis of the proposed algorithm and it shows high specificity of 96% and high sensitivity 96.5% in classifying the human recognition at the multiple view angles.

5.2 Hardware Implementation

The implementation of the proposed algorithm has been carried out by the Python API which is used to extract the features from the GAIT videos and then features are given as the inputs to the FPGA using the UART (Universal Asynchronous Receiver and Transmitter) interfaces. The Xilinx VIVADO version 18 has been used to synthesis the proposed network and ARTIX-7Xc7a0001156 processor has been used for the implementation of fully convolutional requirements that need to be carefully selected for optimal use of resources. BRAM (Block RAM) and DSP blocks are known to be the most important processing and computing resources. The ARTIX-7 FPGA architecture is used for the testing and implementation. Table 5 represents the resource utilization of the proposed pruned algorithm in ARTIX-7 FPGA.

Table 1 and 2 clearly presents the area utilization, power, throughput and delay analysis for the conventional baseline architectures and hardware -software codesign structure for the proposed hybrid algorithms. It is found that the power consumed is 15% reduced than existing architecture and integration of pipelined mechanism in the proposed architecture has reduced the delay by 20% with 22.5% increase in throughput. Table 3 and 4 presents the different area utilization for proposed hybrid architecture using the conventional and hardware-software co-deign methodology. It is clear that conventional(baseline) method of handling the hybrid deep learning models occupies nearly 95% of the total resources which creates the computational overhead process even though it has produced the good performance in classifying the human activities using the CASIA-B video datasets whereas incorporating the hardware-

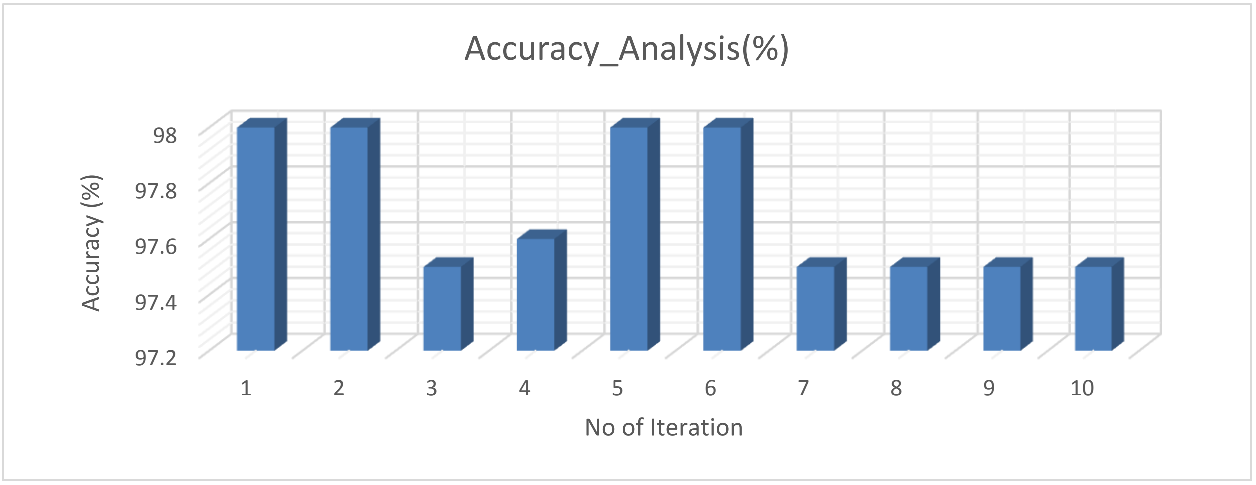

Figure 2 Accuracy Performances of the Proposed Algorithms at 10 Different Iteration(X- Validation Performance)

Recognition Class	Recognition Class										
	0	18	36	54	72	90	108	126	144	162	180
0	3645	12	09	10	11	10	10	10	08	09	10
18	10	3644	11	12	10	10	09	10	09	11	10
36	09	09	3595	06	10	12	10	09	09	10	09
54	08	09	10	3600	05	08	10	08	09	10	12
72	10	09	08	11	3591	10	14	12	08	09	10
90	09	11	08	11	05	3596	10	10	10	09	12
108	09	10	12	11	10	09	3592	10	11	12	12
126	09	11	10	12	11	10	05	3589	10	11	112
144	08	10	11	10	12	11	10	12	3585	10	11
162	09	10	10	10	10	11	10	10	10	3584	10
180	08	09	10	11	12	14	12	12	14	10	**3586**

Figure 3 Confusion matrix for the Proposed Algorithm using the CASIA-B datasets

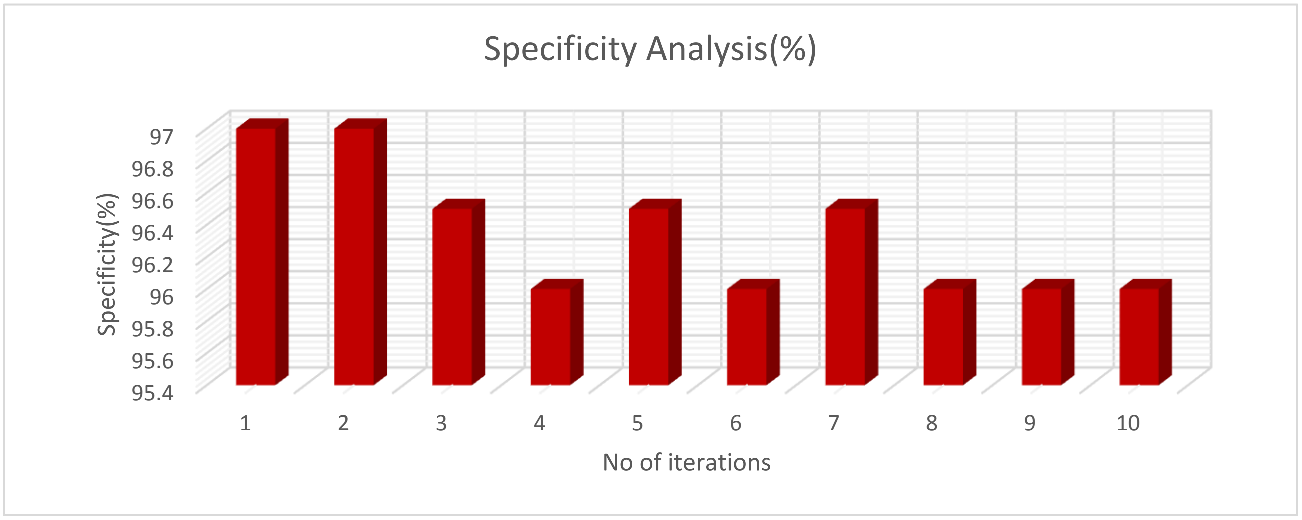

Figure 4 Specificity Performances of the Proposed Algorithms at 10 Different Iteration(X- Validation Performance)

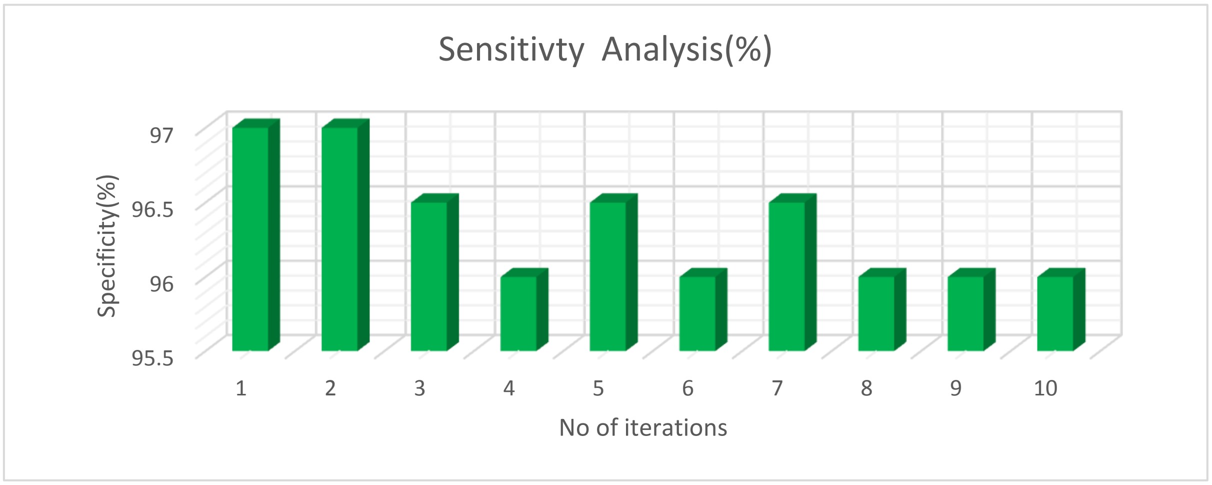

Figure 5 Sensitivity Performances of the Proposed Algorithms at 10 Different Iteration (X- Validation Performance)

Table 1 Power and Area Analysis for the Conventional Method of Implementing the CNN_LSTM networks

SL.no	Model Used	Power(W)	Methodology Incorporated	Area Utilization	Delay	Throughputs GOPS
01	CNN - LSTM	19	Conventional	82%	100.45	67

Table 2 Power and Area Analysis for the Hardware-Software l Method of Implementing the CNN_LSTM **networks**

SL.no	Model Used	Power(W)	Methodology Incorporated	Area Utilization	Delay	Throughputs GOPS
01	CNN - LSTM	15.4	HardwareSoftware Codesign	65%	78.56	78

Table 3 Resource Utilization for Implementing the CNN_LSTM networks using Conventional Methods

SL.no	Model Used	BRAM	LUT	FLIP FLOPS	DSP	FREQUENCY
01	CNN - LSTM	340	12458	289800	735	250 MHz

Table 4 Power and Area Analysis for the Hardware-Software l Method of Implementing the CNN_LSTM networks

SL.no	Model Used	BRAM	LUT	FLIP FLOPS	DSP	FREQUENCY
01	CNN - LSTM	300	12300	278900	645	250 MHz

software design for implementing the hybrid CNN-LSTM models has consumed only 65% of area utilization in which 20% reduction in BRAM, 23% reduction in LUT's, 24% reduction in Flip Flops and 25% reduction in DSP utilization than conventional baseline architectures.

6 CONCLUSION

The paper proposes the FPGA based new hybrid deep learning models for analyzing the different gait videos. The hybrid combination of the CNN and LSTM has been used for the human recognition. The algorithms has been implemented in the ARTIX-7 FPGA and it is found that conventional methodology of implementing the FPGA has created the high computational overhead and more power consumption where as the adoption if hardware-software co-design along with the pipelining architecture has average area reduction by 20%, reduced power consumption by 15%, reduced delay by 20%, higher throughput by 22.5% than the existing baseline architectures. But still improvisation is required in architecture for implementing the hybrid deep learning algorithms for achieving the resource constraints with high performance and low power consumption.

References

[1] Gafurov, D, "A survey of biometric gait recognition: approaches, security and challenges", IEEE Int. Conf. on Biometrics: Theory, Applications and Systems, Norwegian, pp. 1–12, 2007.

[2] Zheng, L., Zhang, Z., Wu, Q., et al, "Enhancing person re-identification by integrating gait biometric", Neurocomputing, Vol. 168, pp. 1144–1156, 2015.

[3] Maodi, H., Wang, Y., Zhang, Z., et al, "Incremental learning for video-based gait recognition with LBP flow", IEEE Trans. Cybern., Vol. 43, no. 1, pp. 77– 89, 2013.

[4] Kusakunniran, W., Wu, Q., Li, H., et al, "Multiple Views gait Recognition using view transformation model based on optimized gait energy image", 12th IEEE Int. Conf. on Computer Vision, pp. 1058–1064, 2009.

[5] Lu, J., Wang, G., Moulin P, "Human identity and gender recognition from gait sequences with arbitrary walking directions", IEEE Trans. Inf. Forensics Sec., Vol. 9, no. 1, pp. 51–61, 2014.

[6]Xiaofan Zhang, Xinheng Liu, Anand Ramachandran, ChuanhaoZhuge, Shibin Tang, Peng Ouyang, Zuofu Cheng, “High-performance video content recognition with long-term recurrent convolutional network for FPGA”, 2017 27th International Conference on Field Programmable Logic and Applications (FPL), 2017.

[7] Kamer Kayaer, Vedat Tavsanoglu, “A new approach to emulate CNN on FPGAs for real time video processing”, 2008 11th International Workshop on Cellular Neural Networks and Their Applications, 2008.

[8] C. Farabet, B. Martini, P. Akse lrod, S. Talay, Y. Le Cun, and E. Culurciello, "Hardware-accelerated convolutional neural networks for synthetic vision systems”, Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010.

[9] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, “A dynamically configurable coprocessor for convolutional neural networks”, ISCA '10: Proceedings of the 37th annual international symposium on Computer architecture, pp. 247–257, 2010.

[10] C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, and Y. LeCun, “Neuflow: A runtime reconfigurable dataflow processor for vision”, CVPR 2011 Workshops, 2011.

[11] N. Li, S. Takaki, Y. Tomioka, and H. Kitazawa, "A multistage dataflow implementation of a deep convolutional neural network based on FPGA for high-speed object recognition", 2016 IEEE South west Symposium on Image Analysis and Interpretation (SSIAI), 2016.

[12] A. Dundar, J. Jin, B. Martini, and E. Culurciello, “Embedded Streaming Deep Neural Networks Accelerator With Applications”, IEEE Trans Neural Netw. And Learn Syst,. Vol. 28, no. 7, pp. 1572-1583, 2017.