Energy Efficient 3D CNN Inference using Multi-dimensional Systolic Architectures on FPGA

Fatima Hameed Khan, Muhammad Adeel Pasha, and Shahid Masud
Department of Electrical Engineering,
Lahore University of Management Sciences (LUMS),
Lahore, Pakistan
Email: {fatima.k, adeel.pasha, smasud}@lums.edu.pk

***Abstract*—The emergence of 3D Convolutional Neural Networks (CNNs) has revolutionized video-related tasks by enabling efficient processing of spatiotemporal data. In this paper, we have proposed a new hardware design for efficient mapping of 3D CNN on 3D systolic array architecture. The architecture enables data reuse across both spatial and temporal dimensions to enhance computational efficiency. Our architecture significantly reduces memory utilization by eliminating the need to store large, replicated data sets typically required for convolution operations. Moreover, we have introduced a generalized dataflow model tailored for 3D systolic architectures, which further increases the throughput. The designed hardware accelerator leverages three-dimensional dataflow to maximize hardware utilization and minimize data transfer latency. An FPGA based implementation of a pre-trained 3D CNN for human activity recognition was carried out. The efficacy of the proposed approach in terms of energy efficiency and latency was evaluated for widely used networks including C3D, I3D, and R(2+1)D. The results demonstrate that our design minimizes the latency and energy dissipation by around 50%.**

Keywords—3D CNNs, Multi-dimensional Dataflow, FPGA, Systolic Architecture

I. INTRODUCTION

Recently, there has been a notable shift in computer vision research towards addressing more complex tasks, particularly those involving the processing of videos. To cater to these emerging requirements, 3D Convolutional Neural Networks (CNNs) are gaining popularity due to their ability to extract features across both spatial and temporal dimensions. The spatiotemporal analysis is beneficial for various tasks such as video action recognition [1-3], medical imaging [4], and volumetric data analysis in fields ranging from surveillance to autonomous driving [5]. However, high computation and memory complexity of 3D CNNs impedes their deployment on resource-constrained and low-power platforms like Field Programmable Gate Arrays (FPGAs). It poses a significant challenge for many edge application where real-time inference and privacy of data is required. The FPGA is preferred over other hardware platforms due to its reconfigurable and flexible architecture and easier path to system development. This paper aims to develop an efficient FPGA-based hardware accelerator to optimize the inference of 3D CNN in terms of latency and energy consumption.

In 3D CNNs, the 3D filter is slid in both spatial as well as temporal dimension to extract the time-domain features. The addition of a temporal dimension aggravates the difficulty to reuse filter and input data in all three dimensions. The convolution operation for each output pixel results in the overlapping of input data in both dimensions. The reported hardware accelerators for 2D CNNs [6] primarily emphasize the spatial reuse of input data by arranging Processing Elements (PE) in a 2D plane. Consequently, this approach lacks efficiency when applied to 3D CNN inference tasks. Several specialized 3D CNN accelerators [7-15] have also been developed to address the issue of temporal reuse. Most of these accelerators decompose the 3D convolution into several 2D convolutions and then process them on different PE arrays. Each PE array utilizes the spatial data locality followed by the transmission of 3D input feature maps to additional on-chip buffers for exploiting temporal data locality. Tian et al. [8] recently presented an accelerator that consists of multiple 2D PE arrays to implement convolution operations using different loop ordering strategies. In [16], Chien and Gupta proposed a design based on multiple systolic arrays, coupled with hierarchical reconfigurable buffers, to leverage in-memory reusability and prevent the need for temporal data re-fetching. These prior architectures lead to additional data transfers between the 2D PE arrays and on-chip memory, resulting in excessive energy costs. The clustering of multiple PE arrays also requires more on-chip resources rendering these un-suitable for low-power and resource-constrained scenarios.

This paper presents the design of an energy-efficient 3D CNN FPGA accelerator that supports data reuse in temporal dimension. The typical 2D systolic array is suited to mapping the complex and overlapped dataflow required for 2D convolution operation. In our work, this idea has been extended for 3D CNNs by expanding the dimension of the systolic array to accommodate the temporal direction. The 3D systolic array is designed to process the sliding of the 3D convolution filter in all the three dimensions without converting it into a matrix form. A corresponding dataflow control logic has also been developed to synchronize the propagation of data in multiple planes of a 3D systolic array. Previously, the authors in [17] presented the concept of a systolic cube. Their dataflow mode is output stationary which has limited data reuse and restricted parallelism. Inferring 3D CNNs on their architecture for large-scale video datasets would necessitate either a significantly large systolic cube or a high number of iterations, resulting in substantial resource utilization or increased latency. Our proposed accelerator enhances the data locality by proposing a dataflow mechanism for weight stationary mode for the 3D systolic array. It can be easily scaled across different levels of parallelism to increase the throughput, as the stationary weights are reused by multiple processing elements simultaneously. Experimental results have proven

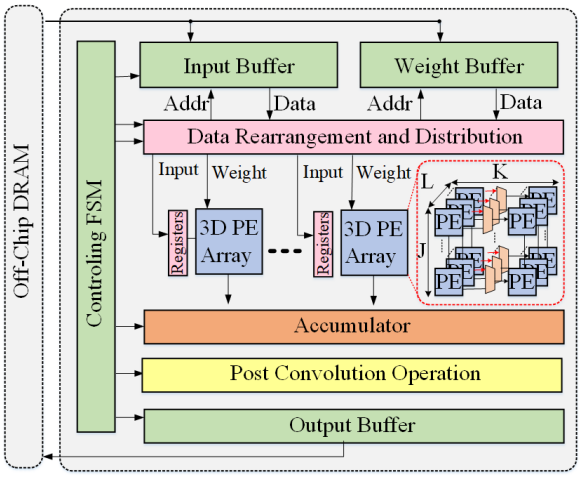

Fig 1. Overall Architecture of the Hardware Accelerator

that the proposed approach significantly enhances efficiency and performance on resource-constrained and low-power hardware. In this context, our paper has three major contributions:

1) An energy efficient 3D systolic architecture for 3D CNNs is proposed that enhances the data reuse across their spatial and temporal dimensions.

2) A generalized dataflow model has been developed for this 3D systolic architecture that maximizes hardware utilization and minimizes the latency in data transfer.

3) The proposed 3D archtitecture has been synthesized and implemented on FPGA as well as Application Specific Integrated Circuit (ASIC) platforms using respective tools.

The rest of the paper is organized as follows. Section II describes the architecture of the proposed hardware accelerator and connections of 3D systolic arrays. Section III presents the overall dataflow model across spatial and temporal dimensions. Section IV includes the experimental setup and evaluation results, and Section V concludes the paper.

II. Hardware Accelerator Design

The proposed 3D systolic architecture fully leverages inter-array data reusability and simplifies memory access patterns caused by data overlapping in 3D convolutions. In this section we have explained the design of an FPGA based hardware accelerator based on the 3D systolic PE array architecture.

A. *Reconfigurable Systolic Array*

In 3D convolution, four-dimensional input ($M \times D_i \times H_i \times W_i$) is convolved with a five-dimensional weigh tensor ($M \times N \times K_D \times K_H \times K_W$) to produce the four-dimensional output feature maps ($N \times D_o \times H_o \times W_o$). Here, spatial dimensions are denoted by $H \times W$, and D represents the temporal depth. M and N are the number of input and output feature maps respectively. $K_D \times K_H \times K_W$ is the size of three-dimensional kernel blocks. Tiling is essential for accelerating this multidimensional data on FPGA due to the limited on-chip memory. Each tile is processed iteratively

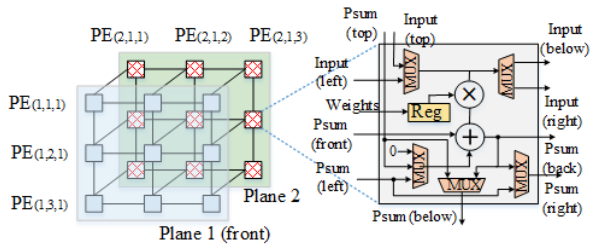

Fig 2. 3D Systolic PE Array Architecture

on our proposed accelerator. Fig. 1 shows the primary component included in the proposed design. The weight buffer stores filter parameters, while inputs and partial outputs are held in the input and output buffers, respectively. The input and weight data are rearranged and distributed to the attached 3D systolic arrays. This arrangement and mapping of data on PE arrays is directed using a PE control logic. The main computation occurs within a 3D PE array module of size $J \times K \times L$ where PEs are interconnected in three dimensions. This 3D systolic architecture loads three-dimensional kernel blocks directly and performs convolution on the incoming input data. The proposed architecture supports the varying kernel sizes across the layers of a 3D CNN by reconfiguring the connection between these 3D PEs arrays. The multiplexers attached to the PEs control the reconfiguration of 3D systolic array. It determines whether a PE receives input from an adjacent PE or directly from the input buffer. The flexibility in PE connections allows the hardware to exploit input and output level parallelism. Kernel blocks that are smaller than $J \times K \times L$ dimension can be computed for multiple output and input feature maps simultaneously to maximize resource utilization.

B. *Multi-Dimensional Systolic Architecture*

The 3D systolic architecture features a unique interconnection pattern for its PEs. Each PE is connected to the PE directly to its right and the PE directly below it, facilitating efficient data flow in the spatial dimension. To manage data flow in the temporal dimension, each 2D plane of PEs is connected to its corresponding PEs in the previous plane. This inter-plane connectivity ensures that temporal data dependencies are efficiently handled. Fig. 2 illustrates the detailed architecture of a PE, highlighting these interconnections and the dataflow paths within the 3D systolic array. Each PE contains a Multiply-Accumulate (MAC) unit, associated control logic, data register, and multiplexers. The movement of input data and partial sum (Psum) is controlled in all connected dimensions of PE. Each plane of PEs array receives different inputs from the front side, and a group of inputs is passed to registers and shared by a column of PEs in each cycle. Once the input data is fetched from the buffer, it is fully utilized across both spatial and temporal dimensions through the 3D interconnection of PEs. This ensures maximum data reuse and efficient computation within the systolic array. Activation and other post-convolutional operations are performed at the end of the pipelining stage. The next section explains the detailed dataflow to regulate the synchronous propagation of data within the 3D systolic array.

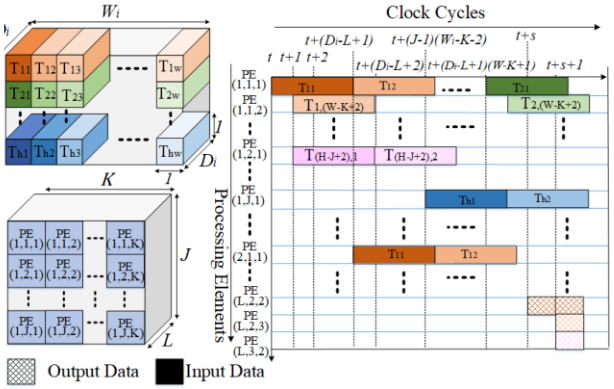

Fig 3. Generalized Dataflow for 3D Systolic Architecture

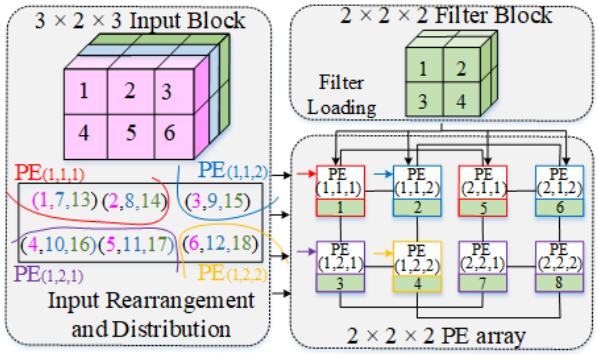

Fig 4. Mapping of 3D Convolution on 3D PE Array

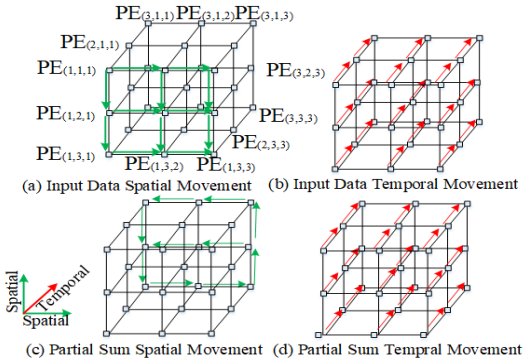

Fig 5. Dataflow for 3D Convolution on 3D Systolic Array (a) Input Data Movement (b) Partial Sum Movement

III. Dataflow for 3D Systolic Architecture

The traditional 2D systolic array utilizes spatial data localities but struggles with data overlapping in the temporal dimension. The ability of the hardware to exploit various types of data locality depends on the architecture design and the dataflow. This section describes the dataflow within the 3D systolic architecture to fully exploit the data reusability and to reduce the frequency of data requests from on-chip buffers. The weights are loaded into the systolic array in their existing arrangement as the weight stationary mode is adopted for the proposed dataflow. The overall flow of the input data and the generation of the output block is shown in Fig. 3. The input data is rearranged along the temporal dimension. As a result, $H \times W$ temporal blocks are formed, as illustrated in Fig. 3. All these temporal blocks are fed to the 3D PE array through the front plane of the array. Each PE in the front plane will receive the $T_{a,b}$ block of the input data. The terms a and b are defined for each PE as follows:

$$a = 1{:}H - (K_H - 1), b = 1{:}W - (K_W - 1) \rightarrow PE_{1,1,1} \quad (1)$$

$$a = H - K_H + j, \qquad b = 1{:}W - (K_W - 1) \rightarrow PE_{1,j,1}$$
$$where\ j \in 2, 3, \dots, J \quad (2)$$

$$a = 1{:}H - (K_H - 1), b = W - K_W + k \rightarrow PE_{1,1,k}$$
$$where\ k \in 2, 3, \dots, K \quad (3)$$

$$a = H - (K_H - j), b = W - (K_W - k) \rightarrow PE_{1,j,k}$$
$$where\ j \in 2, 3, \dots, J\ and\ k \in 2, 3, \dots, K \quad (4)$$

Here, it is assumed that J,$K$and L are exactly equal to K_D, K_H and K_W respectively. The input data block will arrive at each PE in a systolic manner and the arrival of temporal blocks at each PE block is determined by the size of the filter and input block, as formulated in Equations (1) – (4). In multi-dimensional systolic architecture, there is a conditional data movement in each direction. The input block will be passed to its previous plane after every $D - K_D + 1$ clock cycles. The spatial moment of the input data depends on the arrival of the new data block. When a new data block arrives at any PE, that PE does not receive the data from its top and left neighboring PEs. However, it transfers the incoming data to its neighboring PEs. As an example, the input block of $3 \times 2 \times 3$ is mapped to $2 \times 2 \times 2$ PE array to convolve with $2 \times 2 \times 2$ filter block. Fig. 4 shows the input data rearrangements in temporal blocks and the distribution of these blocks for each PE. The filter block is loaded in their original arrangement onto the PE array. The dataflow for 3D systolic architecture is illustrated in Fig. 5. As shown in Fig. 5(a), the input data is loaded from the front plane only, which is shared in both temporal and spatial dimensions. The previous planes carry the temporal movement only, as shown in Fig. 5(b). The movement of each partial sum is highlighted in Fig. 5(c) and 5(d). In this case, the front plane share the calculated partial sum in the temporal dimension only whereas the spatial accumulation occurs in the last plane. Table I shows the computation in each PE block for the given example. Every PE in the last plane generates the final output pixel and the order of the output is the same as that of the input temporal block. Using this dataflow, any 3D CNN can be mapped on a 3D systolic array while exploiting maximum data reuse. It takes only 13 clock cycles to compute the output data whereas the work in [17] takes 20 clock cycles for the same example. Hence, the proposed 3D systolic architecture reduces the on-chip memory access and enhances the performance of 3D CNN acceleration.

IV. Results and Discussion

The Xilinx Virtex-7 FPGA VC709 was chosen as the implementation platform. It features a Virtex-7 690T FPGA

Table I. Computation in each PE for Every Clock Cycle

clk no.	PE (111)	PE (112)	PE (121)	PE (122)	PE (211)	PE (212)	PE (221)	PE (222)
1	1×1							
3	7×1	3×2	4×3		7×5			
5	2×1	9×2	10×3	6×4	13×5	9×6	10×7	
7	8×1	2×2	5×3	12×4	8×5	15×6	16×7	12×8
9		8×2	11×3	5×4	14×5	8×6	11×7	18×8
11				11×4		14×3	17×7	11×8
13								17×8

TABLE II. COMPARISON OF FPGA IMPLEMENTATION PARAMETERS WITH THE EXISTING WORK

Architecture	[10]	[11]	[8]	[13]		[7]	[14]		[15]	Proposed Work		
Model	C3D	C3D	C3D	C3D	R(2+1D	I3D	C3D	R(2+)D	X3D	C3D	I3D	R(2+1)D
Throughput (GOP/s)	172	334	357	79	35	1145	424	185	119	1024	1898	684
Latency(ms)	35.3	115	107	487	243	96	91	46	53	73	69	35
Energy/clip(J)	1.9	1.8	-	3.2	1.6	1.4	1.72	1.72	1.4	1.02	0.95	0.93
DSP(%)	93	99	96	48	48	98	98	98	86	0	0	0
LUT(%)	45	62	-	54	54	85	62	62	-	59	59	59
BRAM(%)	-	26	52	100	100	79	63	65	-	62	62	62

and two 4GB DDR3 DRAMs operating at 200 MHz. The design is implemented using the Vivado Design Suite (20.1), and all results are reported post place-and-route. For ASIC implementation, the area for the proposed hardware architecture is determined using the Cadence Genus synthesis tool for 180 nm technology. The experiments were conducted on popular 3D networks, including I3D, C3D, and R(2+1)D, using the UCF101 dataset. These models are inferred using 8-bit precision. A detailed analysis of the proposed methodology is based on two main factors: latency and energy estimation. The energy is estimated using the analytical model presented in [18]. The tiling size configuration for each model is obtained through experimentation.

The proposed 3D systolic hardware is compared below with the classical 2D systolic array and the systolic cube architecture [17]. Our 3D PE array consists of 9 rows, 9 columns, and 9 planes to accommodate all kernel sizes of the considered 3D networks. The baseline 2D array is at the same scale, with 9 rows and 9 columns and 9 such arrays are used for a fair comparison in throughput and resource consumption. The systolic cube configuration is also $9 \times 9 \times 9$ but with an output stationary dataflow as mentioned earlier in Section II. Fig. 6 shows the latency based performance comparison of different architectures normalized to the classic 2D array implementation, measured in execution cycles. Given that 3D convolutions are implemented through multiple 2D convolutions in traditional 2D accelerators, only parts of the data locality are utilized in 2D-plane accelerators. However, in the 3D systolic cube, weights and inputs can be shared across neighboring PE planes. The networks, C3D and I3D have network structures with sufficient temporal depth and feature channels typically aligned to multiples of eight, contributing to higher hardware utilization. Compared to 2D systolic array and systolic cube architectures, the proposed hardware achieves 6.4× and 1.92× performance speedup on average, respectively.

We have estimated the energy consumption of the three 3D CNN networks on these systolic hardware

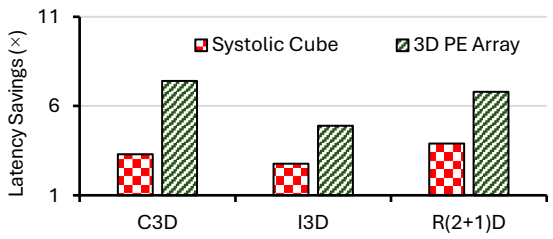

Fig 6. Relative Latency Savings of Systolic Cube and the Proposed 3D systolic PE Array Architecture against the Traditional 2D Arrays

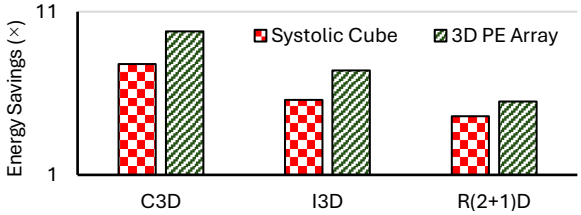

Fig 7. Relative Latency Savings of Systolic Cube and the Proposed 3D systolic PE Array Architecture against the Traditional 2D Arrays

implementations. The energy reduction in the systolic cube and 3D systolic array design is primarily due to better data locality in both the temporal and spatial dimensions, significantly reducing buffer accesses. The energy efficiency analysis, demonstrated in Fig. 7, shows that the 3D systolic array architecture offers more energy savings compared the state-of-the-art approaches, perfectly aligning with the design goals for an embedded application. The ASIC implementation of the proposed hardware design is also carried out for a fair comparison with [17] as their work targets an ASIC platform. We have achieved an area efficiency of 0.74 and 0.70 FPS/mm^2 for I3D and C3D models, respectively that maps to a relative improvement of 9.1× and 2.8× in comparison with the work reported in [17].

Table II provides a detailed comparison with the previous implementation. The proposed 3D systolic cube architecture significantly enhances the efficiency of 3D CNN computations by leveraging superior data locality in both temporal and spatial dimensions. This design minimizes the latency by 44%, 29%, and 55% while saving on average 49% 33%, and 45% of the energy for C3D, I3D and R(2+1)D, respectively. Through comprehensive experiments on these popular 3D CNN networks, we have demonstrated that our proposed approach achieves significant speedup and energy savings compared to traditional 2D systolic architectures.

V. CONCLUSION

The proposed 3D systolic architecture significantly enhances the efficiency and performance of 3D CNNs by optimizing data reuse across all dimensions. Our design addresses the limitations of traditional 2D systolic array architectures by reducing their data transfer requirements. The integration of a generalized dataflow model further maximizes hardware utilization, enabling effective acceleration of computationally intensive 3D CNNs on resource-constrained and low-power hardware platforms. Experimental results on popular networks such as C3D, I3D, and R(2+1)D demonstrate the superiority of our approach in terms of throughput and energy efficiency. The results reported from FPGA and ASIC implementation corroborate the advantages of our proposed approach.

REFERENCES

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks", In IEEE International Conference on Computer Vision (ICCV), pp. 4489-4497, 2015.

[2] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724-4733, 2017.

[3] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition", In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6450-6459 2018.

[4] H. Zunair, A. Rahman, N. Mohammed and J. Cohen, "Uniformizing Techniques to Process CT scans with 3D CNNs for Tuberculosis Prediction," International Workshop on Predictive Intelligence In Medicine, Springer, pp. 156-168, 2020.

[5] S. Mittal and Vibhu, "A Survey of Accelerator Architectures for 3D Convolution Neural Networks," Journal of Systems Architecture, vol. 115, p. 102041, 2022.

[6] K. Khalil, A. Kumar and M. Bayoumi, "Low-Power Convolutional Neural Network Accelerator on FPGA," IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hangzhou, China, pp. 1-5, 2023.

[7] F. H. Khan, M. A. Pasha and S. Masud, "Towards Designing a Hardware Accelerator for 3D Convolutional Neural Networks", Computers and Electrical Engineering,, vol. 105, pp. 108489,2023.

[8] T. Tian, X. Jin, L. Zhao, X. Wang, J. Wang and W. Wu, "Exploration of Memory Access Optimization for FPGA-based 3D CNN Accelerator," Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2020, pp. 1650-1655, 2020.

[9] Y. Yang, W. He, J. Hu, S. Cheng and W. Xu, "FGPA-Based Design for Accelerating 3D Convolutional Neural Network", International Journal of Frontiers in Engineering Technology, vol. 5(3), pp. 40-48, 2023.

[10] H. Fan et al., "F-E3D: FPGA-based Acceleration of an Efficient 3D Convolutional Neural Network for Human Action Recognition," IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP), New York, NY, USA, pp. 1-8, 2019.

[11] Z. Liu, P. Chow, J. Xu, J. Jiang, Y. Dou and J. Zhou, "A Uniform Architecture Design for Accelerating 2D and 3D CNNs on FPGAs", *Electronics MDPI*, vol. 8, no. 1, 1 2019.

[12] J. Shen, Y. Huang, Z. Wang, Y. Qiao, M. Wen and C. Zhang, "Towards a Uniform Template-based Architecture for Accelerating 2D and 3D CNNs on FPGA", *Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, vol. 8(1) pp. 97-106, 2018.

[13] M. Sun, P. Zhao, M. Gungor, M. Pedram, M. Leeser and X. Lin, "3D CNN Acceleration on FPGA using Hardware-Aware Pruning," *57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2020, pp. 1-6,2020.*

[14] P. Toupas, A. Montgomerie-corcoran, C.-s. Bouganis and D. Tzovaras, "Harflow3d: A Latency-Oriented 3D-CNN Accelerator Toolflow for HAR on FPGA Devices", *Proceedings of International Symposium on Field-Programmable Custom Computing Machines FCCM*, 2023.

[15] P. Toupas, C. -S. Bouganis and D. Tzovaras, "FMM-X3D: FPGA-Based Modeling and Mapping of X3D for Human Action Recognition," IEEE 34th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Porto, Portugal, pp. 119-126, 2023.

[16] A. A. Chien and R. K. Gupta, "MORPH: a system architecture for robust high performance using customization (an NSF 100 TeraOps point design study)," *Proceedings of 6th Symposium on the Frontiers of Massively Parallel Computation (Frontiers '96)*, Annapolis, MD, USA, 1996, pp. 336-345, 2019.

[17] Y. Wang, Y. Wang, C. Shi, L. Cheng, H. Li and X. Li, "An Edge 3D CNN Accelerator for Low-Power Activity Recognition," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 5, pp. 918-930, May 2021.

[18] Y. -H. Chen, T. Krishna, J. S. Emer and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017.