Dimension fusion: Dimension-level dynamically composable accelerator for convolutional neural networks

Huipeng Deng¹, *, a), Jian Wang¹, *, Huafeng Ye², Shanlin Xiao², Xiangyu Meng¹, and Zhiyi Yu²

Abstract Convolutional neural networks (CNNs) have proven to be promising in various applications such as audio recognition, image classification, and video understanding. Winograd algorithm helps to reduce the complexity of computation in a convolution but suffers from poor compatibility for different convolution shapes. This work introduces a dynamic dimension-level fusion architecture based on Winograd for accelerating different dimensions of CNNs. We explore this Winograd architecture by designing Dimension Fusion, a dimension-level processing engine that dynamically fuses to match the convolution shape of individual CNN layers. The proposed architecture is the first work based on Winograd algorithm to be compatible with all convolution shapes (dimension, stride, and filter-size) and achieves highest PE efficiency up to 1.55x and energy efficiency up to 3.3x compared with the state-of-art accelerators.

Keywords: dynamic fusion, Winograd algorithm, accelerators, convolutional neural networks

Classification: Integrated circuits (memory, logic, analog, RF; sensor)

1. Introduction

Deep convolutional neural networks (CNNs) architecture achieves state-of-the-art performance in a variety of interdisciplinary researches, including speech recognition [1], image classification [2, 3], and video understanding [4, 5, 6]. In order to deal with multi-dimensional data such as acoustic signals, images, or videos, 1D/2D/3D convolutional neural networks (CNNs) architectures are proposed. The single signal may limit the performance capacity of CNNs. Several works [7, 8, 9] manage to exploit multiple signals to improve the recognition performance in medical and video. Besides, previous researches [10, 11] have focused on mixed dimensional convolution ((2+1)D) to alleviate the computation burden of deep CNNs. Fig. 1 shows the applications of different structured data of CNNs. Abundant studies [12, 13, 14] have focused on accelerating specific dimensional CNNs. However, to adapt to various application scenarios, mobile devices have to handle a variety of dimensional CNNs.

Winograd algorithm has shown dramatic improvement in the CNN performance by reducing the computation complexity of convolution. However, the less flexibility of current Winograd implementations [15, 16, 17, 18, 19, 20] makes it challenging to accelerate different CNNs (different dimensions, strides, and filter sizes) efficiently. Thus, a Winograd-based accelerator would either incur significant waste of computational resources to accommodate different convolution shapes or lead to limited support for convolution. Besides, although the computation patterns of different dimensional CNNs are similar, the computational complexity, network mapping methods, and data reuse vary greatly. Fully realizing the optimization of flexibility and performance is a challenge for designing a general-purpose Winograd CNN accelerator.

To accelerate the execution of different shapes of CNNs for various application scenarios, this work contributes a novel Winograd acceleration architecture, called Dimension Fusion that overcomes the shortcomings of the previous Winograd implementations in that significant adaptability for various architectures (different dimensions, strides, and filter sizes) can be achieved on the hardware prototypes. Besides, a novel dimension fusion/composition method is proposed, which dynamically matches the dimension-level processing engines to the varying convolution shapes required by CNN layers. By offering this flexibility, the dimension-level flexible Winograd architecture, Dimension Fusion, aims to minimize the complexity of computation and accelerating all shapes of convolution. We also implement the proposed architecture in the Xilinx FPGA ZCU102 and 65 nm CMOS technology. Compared with the state-of-art accelerators, the proposed accelerator is compatible with all convolution shapes and achieves highest PE efficiency up to 1.55x and energy efficiency up to 3.3x.

2. Dimension fusion approach

2.1 Winograd algorithm

Winograd [21] proposed an efficient algorithm to reduce the arithmetic complexity of convolutional operations. Equations (1)–(3) show the Winograd algorithms of different di-
mensions (1D, 2D, and 3D), where ⊙ is denoted as element-wise multiplication. G, B, A are constant transformation matrices with simple values for output, input and filter, which means the transformation only need to perform addition and subtraction, but not multiplications. To simplify the discussion, we denote a 1D Winograd algorithms as F(m, r) which the filter size is r and the partial sum size is m. Besides, the F(mxm, rxr) and F(mxmxm, rxr) are used to represent the 2D, and 3D Winograd algorithms. The transformation matrices (G, B, and A) are determined for a given m and r. Compared to regular convolution, 1D/2D/3D Winograd algorithm makes it possible to support different dimension CNNs in a Winograd accelerator.

2.2 Dimension fusion

Our previous work [20] used a 2D/3D fusion unit to process the 2D/3D convolution layers. The drawbacks of this methodology include the poor compatibility for 1D convolution and the under-utilization of the fusion unit. Using 2D Winograd unit to accelerate 1D convolution brings invalid calculation and causes low PE efficiency. For example, the 1D convolution (the filter size is 3 and partial sum size is 2) operated in 2D Winograd unit F(2,3) still needs 16 multiplication operations when the regular convolution needs 6 multiplication operations. To alleviate these deficiencies, we proposed the fusion unit (Dimension-level processing engine) supports both 1D/2D/3D Winograd algorithms and different Winograd shapes (F(2, 3) and F(3, 2)) and a dynamic dimension-level fusion/decomposition approach to match all the convolution shapes. Fig. 2 shows the dimension fusion/composition method which helps different convolution shapes implement efficiently in our fusion unit. The filter is decomposed by the relationship of the data, and the data with the same position will be combined into one type of our fusion unit. Once there are not enough data to combined into a unit, 0 needs to be filled. The relationships of data depend on the stride and filter size of convolution. The above fusion/composition method also can be easily extended to a broader case when the filter size and stride are larger. As Fig.2 shows, the filter of different size is decomposed into 1D/2D/3D convolution. Moreover, previous approaches to deal with the large stride and large filter size.

Reference [17, 19] introduced the universal approach to deal with the large stride and large filter size. Previous design space exploration schemes have been applicable only to 2D CNN accelerators, making them unsuitable for different dimension architecture. Although the sizes of the design space of 1D/2D/3D dimension CNN accelerators are different, the similarity of different dimension Winograd algorithm makes it possible to support different dimension CNNs in a Winograd accelerator.

\[
y = G[(Bx)⊙(Aw)]
\] (1)
\[
y = G((Bx B^T)⊙(Aw A^T)) G^T
\] (2)
\[
y = (G((Bx B^T R) B^T)⊙((Aw A^T R) A^T) G^T) R^T G^T
\] (3)
\[
B = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & -1 & -1 \\
-1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad A = \begin{bmatrix}
1 & 0 & 0 \\
1/2 & 1/2 & 1/2 \\
0 & 1/2 & -1/2 & 1/2 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \\
G = \begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 1 & -1 & 1 \\
\end{bmatrix}
\] (4)
\[
B = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & -1 & -1 \\
-1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad A = \begin{bmatrix}
1 & 0 & 0 \\
1/2 & 1/2 & 1/2 \\
0 & 1/2 & -1/2 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \\
G = \begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 1 & -1 & 0 \\
0 & 1 & 1 & 1 \\
\end{bmatrix}
\] (5)

Although the Winograd algorithm helps to reduce the complexity of computation in a convolution, it also suffers from the complication of transformation matrices and poor compatibility for different convolution shapes. When the filter size r increase, the parameters of transformation matrices become more complicated, which makes it hard to implement in hardware and causes non-negligible accuracy loss. Moreover, the small filter size is commonly used in CNN models. Therefore, a small Winograd computation unit is the most attractive alternative. Equations (4), (5) are for F(2, 3) and F(3, 2). Since the transformation matrices of F(2, 3) and F(3, 2) have high similarity in many same operators, the basic dimension computation unit of our accelerator is F(2, 3) and F(3, 2).

On the other hand, the certain Winograd F(m, r) computation unit can only accelerate the convolution with filter size r and stride 1 efficiently. At present, many works [14, 18] based on the Winograd algorithm only focus on 1-stride convolutions. Several works manage to exploit the compatibility for 2D convolution sizes (large stride, large filter size). WHD [16] exploited the fusion of Winograd unit, but only one type of unit can be used in per-layer. Reference [22] proposed a Winograd processing element for convolutions which only compatible with filter size 3 and both stride 1 and 2. Reference [17, 19] introduced the universal approach to deal with the large stride and large filter size. Previous design space exploration schemes have been applicable only to 2D CNN accelerators, making them unsuitable for different dimension architecture. Although the sizes of the design space of 1D/2D/3D dimension CNN accelerators are different, the similarity of different dimension Winograd algorithm makes it possible to support different dimension CNNs in a Winograd accelerator.
CNNs [10, 11] have focused on low dimensional convolution (1D) to alleviate the computation burden of deep CNNs. Therefore, it’s necessary to accelerate 1D/2D/3D convolution efficiently.

Because of the high flexibility of the Dimension-level processing engine, the data can be decomposed and combined into the most suitable Winograd unit. Therefore, our fusion/decomposition method achieves the higher efficiency of computational resources with fewer zeros filled. The PE efficiency is (the multiplication operations in regular convolution)/(the multiplication operations in Winograd unit).

Fig. 3 shows the PE efficiency of the processing engine for different convolution shapes. Respectively, the PE efficiency of the processing engine is up to 3.37x compared with convolution and 1.6x compared with WHD [16] which means the processing engine achieves higher performance with negligible transform operations. To minimize the complexity of computation and maximize the utilization of the computation unit, Dimension Fusion dynamically matches the convolution shape (dimension, stride, filter size) required for the CNN, which may vary layer by layer, without any loss inaccuracy.

3. Dimension fusion architecture

In this section, the methodology of architecture design of Dimension Fusion is given. First, with the flexible control logic, the Dimension Fusion is achieved from two levels: transformation matrices fusion and dimension-level unit fusion. Second, SRAM and register accesses dominate the energy consumption when accelerating CNNs. Therefore, a hybrid dimensional buffer is given to maximize data reuse and reduce the off-chip bandwidth. Finally, CNN provides different degrees of data-level parallelism in various dimension. Therefore, dimension fusion systolic array is given to minimizing the overhead of dynamically constructing Dimension-Engines and minimizes the overhead of transformation matrices in the accelerator.

3.1 Dimension-level processing engine

Dimension fusion is a collection of dimension-level fusion units and fusion transformation matrices (FTM), that dynamically compose to logically construct dimension-level processing engine that executes Winograd operations with the required shapes. Dimension Fusion arranges the fusion unit and transformation matrices in a 3-dimensional physical grouping, called Dimension-Engines. Each 1D fusion unit (F-1D) can perform individual 1D Winograd operations F(2, 3) or F(3, 2). And fusion units can be fused logically to perform different dimensional Winograd operations. As Fig. 4 shows, four fusion units logically fuse to form 2D-fusion unit (F-2D) which performs the 2D Winograd operations F(2x2, 3x3) or F(3x3, 2x2). After a certain logical rotation, four 2D-fusion unit fuse to form 3D-fusion unit (F-3D) that enables 3D Winograd operations F(2x2x2, 3x3x3) and F(3x3x3, 2x2x2).

For 1D Winograd operations (Fig. 4(b)), each Dimension-Engine contains 16 1D-fusion units, offering the highest parallelism. The parallelism of the input channel and output channel is 4. Therefore, Dimension-Engine adds the results from 4 1D-fusion units to generate a single outgoing partial sum (size is 2 or 3). For 2D Winograd operations (Fig. 4(c)), each 2D-fusion unit contains four 1D-fusion units and each Dimension-Engine contains 4 2D-fusion units. The parallelism of the input channel is 4 and the output channel is 1. Therefore, Dimension-Engine adds the results from 4 2D-fusion units to generate a single outgoing partial sum (size is 2x2 or 3x3). For 3D Winograd operations (Fig. 4(d)), each Dimension-Engine contains a 3D-fusion units. And a single outgoing partial sum (size is 2x2x2 or 3x3x3) will be generated each cycle.

The dimension of operation supported by Dimension-Engine depends on the spatial arrangement of fusion units fused together and the dynamic arrangement of the transformation matrices. Fig. 5 shows the arrangement of the transformation matrices [23]. Three paths represent three dimensional operations. As the equation (4, 5) shows, the transformation matrices can be efficiently implemented on hardware by using simple arithmetic operations (e.g., ADD and SUB). Furthermore, the fusion transformation matrices of F(2, 3) and F(3, 2) have high similarity in many same operators. The evaluation shows the fusion transformation
matrices only utilize 5% and 12% LUTs more than the matrices of F(2, 3) and F(3, 2).

Finally, the fusion transformation matrices and dimension-level fusion units dynamically form to the Dimension-Engine that can operate on 1D/2D/3D Winograd operations. The above-mentioned fusion/composition method decides the computation type of combined data and the Dimension-Engine is combined into corresponding Winograd units logically according to computation type of data. That is, the Dimension-Engine with the above fusion/composition method enables the architecture to expose the maximum possible level of parallelism with the finest granularity that matches all the convolution shapes (stride, filter size, dimension).

3.2 Hybrid dimensional buffer

Depending on the process mode of Dimension-Engine and data access patterns of different convolution shapes, the buffers must supply different numbers of operation with various dimension. As such, we propose the hybrid dimensional buffer (HDB) to deal with complicated data access patterns by the flexible control logic. Moreover, to avoid redundant accesses to the data array of the buffer which conserves energy, the hybrid dimensional buffer contains the address manage unit (AMU) which controls the data reuse and read/write patterns when deploying different operation shapes. AMU uses the configurable finite state machine which is programmed by registers to generate the read/write address of buffer. Three sets of configurable registers are adopted to denote row loop, column loop, and frame loop for a parameterizable data fetch which is determined at computing time. The hybrid dimensional buffer is a 4-dimensional (row, column, frame, depth) physical grouping. Fig. 6 shows 3-dimensional (row, frame, depth) physical grouping. Fig. 6 illustrates the data reuse method of row data when the operation pattern is F(2, 3) and F(3, 2). The method can be easily extended to column data reuse and frame data reuse. Benefit from the flexible address control of AMU, the hybrid dimensional buffer doesn’t store redundant data and achieves high data reuse.

3.3 Dimension fusion systolic array

To utilize the data-level parallelism in various dimension, we employ a novel 2-dimensional systolic array of fusion units as the architecture for Dimension Fusion. Fig. 7 shows the system-level architecture of the proposed Dimension fusion accelerator. The organization of systolic array provides high-level data reuse to minimize access to on-chip memory which alleviates the requirement for buffer bandwidth. Moreover, systolic execution minimizing the overhead of dynamically constructing Dimension-Engines and minimize the overhead of fusion transformation matrices. As depicted in Fig. 7, the systolic array shares one Activation FTM group which provides 1D/2D/3D data transformation. Dimension fusion (DF) controller is proposed to achieve the above-mentioned dimension fusion/composition method. The Psum HDB is a one-cycle accumulation HDB. As shown in Fig. 7, the Psum HDB resides on the right and collects the psum which is the sum of flowing results and the pre-read result. Finally, the entire systolic array composed of fusion unit provides high expansibility and flexibility for system which supports all shapes of convolution (dimension, filter size, stride).

4. Implementation results

4.1 Layer-wise performance evaluation

Dimension Fusion aims to accelerate the inference of a wide range of CNN models with varying convolution shapes requirements based on the Winograd algorithm. Therefore, We compare the throughput of Dimension Fusion and the
other researches with different shapes of convolution layers that have dimension ranging from 1D to 3D, different strides, and different filter sizes. To make a fair comparison with the researches [16, 20, 28], we implement these approaches with the same hardware resources (256 DSPs) in FPGA ZCU102 and run at a clock frequency of 200 MHz. The specific convolution layer parameter comes from the mainstream CNNs [11, 24, 25, 26, 27]. The result shows that Dimension Fusion not only supports various CNN models but also achieves higher performances (a 1.31 to 4.06 times improvement in precision) compared with other FPGA-based accelerators [18, 20].

4.2 Performance comparison to other CNN accelerators
Dimension Fusion has implemented in TSMC 65 nm technology and ZCU102 FPGA. The Dimension Fusion (ASIC) was placed-and-routed and passed DRC and LVS, the results are from post-layout cycle-accurate gate-level simulations with switching activities profiled from running the actual weights of the C3D and VGG-16 and data from the UCF-101 dataset [29] and ImageNet dataset [30]. Fig. 8 shows the layout of Dimension Fusion. As Table II shows, compared with the state-of-the-art prior ASIC-based accelerators [31, 32], Dimension Fusion (ASIC) achieves highest energy efficiency up to 3.3x. Table III shows that Dimension Fusion (FPGA) achieves highest DSP efficiency up to 1.55x compared with other FPGA-based accelerators [18, 20].

Table II Comparison with other ASIC-based accelerators.

| Platform | JSSC [31] | TCASII [32] | This work |
|----------|-----------|-------------|------------|
| Process  | ASIC      | ASIC        | ASIC       |
| Network  | NA        | NA          | 256 DSPs   |
| Voltage  | 1.2 V     | 1.2 V       | 1.2 V      |
| Clock Frequency | 200 MHz   | 177 MHz     | 310 MHz    |
| Precision (bit) | 1.16      | 9           | 8          |
| DSPP(PE)  | 13824(bit serial) | 256      | 256        |
| Area (mm²) | 16        | 9.83        | 8.7        |
| Performance (GOPS) | 345.6    | 109.2       | 425/319    |
| Power (W) | 0.297     | 0.254       | 0.3        |
| Energy Efficiency (GOPS/W) | 1163   | 429         | 1416/1063  |

Table III Comparison with other FPGA-based accelerators.

| Platform | FCCM21 [20] | TCAD20 [18] | This work |
|----------|-------------|-------------|------------|
| Network  | ZCU102      | VC3/118     | ZCU102     |
| Clock Frequency | 200 MHz   | 200 MHz     | 300 MHz    |
| Precision (bit) | 8         | 16          | 8          |
| DSP  | 1024       | 5184        | 256        |
| LUT  | 192K       | 560K        | 56K        |
| Brains (18 Kb) | 712       | 3392        | 100        |
| Performance (GOPS) | 1353    | 5054        | 387/290    |
| Power (W) | 10.2      | 32          | 1.6        |
| DSP Efficiency  | 1.32      | 0.97        | 1.51/1.13  |
| Energy Efficiency (GOPS/W) | 135    | 157.9       | 241.8/181  |

5. Conclusion
Winograd algorithm helps to reduce the complexity of computation in a convolution but suffers from poor compatibility for different convolution shapes. Therefore, we propose Dimension Fusion, a dimension-level dynamically composable architecture based on Winograd, for their flexible and efficient acceleration. This is the first work that proposed a general-purpose Winograd accelerator which supports all convolution shapes. Dimension Fusion is implemented on the Xilinx ZCU102 FPGA and TSMC 65 nm technology. The evaluation results show that Dimension Fusion achieves highest DSP efficiency up to 1.55x and energy efficiency up to 3.3x compared with the state-of-art accelerators.

Acknowledgments
This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2202601 and Grant 2017YFA0206200; in part by the National Natural Science Foundation of China (NSFC) under Grant 61834005 and Grant 61902443. (Corresponding authors: Zhiyi Yu; Shanlin Xiao).

References
[1] P. Zhou, et al.: “State-clustering based multiple deep neural networks modeling approach for speech recognition,” IEEE/ACM Trans. Audio, Speech, Language Process. 23 (2015) 631 (DOI: 10.1109/TASLP.2015.2392944).
[2] A. DiSpirito, et al.: “Reconstructing undersampled photoacoustic microscopy images using deep learning,” IEEE Trans. Med. Imag. 40 (2021) 562 (DOI: 10.1109/TMI.2020.3031541).
[3] H. Shin, et al.: “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” IEEE Trans. Med. Imag. 35 (2016) 1285 (DOI: 10.1109/TMI.2016.2528162).
[4] A.H. Fan, et al.: “F-C3D: FPGA-based 3-dimensional convolutional neural network,” 2017 27th International Conference on Field Programmable Logic and Applications (2017) 1 (DOI: 10.23919/FPL.2017.8056779).
[5] Y. Wang, et al.: “An edge 3D CNN accelerator for low-power activity recognition,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 40 (2021) 918 (DOI: 10.1109/TCAD.2020.3011042).
[6] K. Hegde, et al.: “Morph: flexible acceleration for 3D CNN-based video understanding,” 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (2018) 933 (DOI: 10.1109/MICRO.2018.00080).
[7] H. Wu, et al.: “Spatiotemporal multimodal learning with 3D CNNs for
A. Krizhevsky, K. He, J. Wang, J. Yepez and S. Ko: “Stride 2 1-D, 2-D, and 3-D Winograd for
A. Lavin and S. Gray: “Fast algorithms for convolutional neural networks,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 28 (2020) 853 (DOI: 10.1109/TVLSI.2019.2961602).

J. Wang, et al.: “FWUA: a flexible winograd-based uniform accelerator for 1D/2D/3D CNNs,” submitted to The 4th IEEE International Conference on Integrated Circuits, Technologies and Applications.

K. He, et al.: “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 770 (DOI: 10.1109/CVPR.2016.9218571).

K. Soomro, et al.: “UCF101: a dataset of 101 human action classes from videos in the wild,” CRCV-TR-12-01 (2012).

O. Russakovsky, et al.: “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis. 115 (2015) 211 (DOI: 10.1007/s11263-015-0816-y).

J. Lee, et al.: “UNPU: an energy-efficient deep neural network accelerator with fully variable weight bit precision,” IEEE J. Solid-State Circuits 54 (2019) 173 (DOI: 10.1109/JSSC.2018.2865489).

L. Rao, et al.: “An energy-efficient accelerator for rain removal based on convolutional neural network,” IEEE Trans. Circuits Syst. II, Exp. Briefs 68 (2021) 2957 (DOI: 10.1109/TCSII.2021.3071455).

C. Yang, et al.: “WRA: a 2.2-to-6.3 TOPS highly unified dynamically reconfigurable accelerator using a novel winograd decomposition algorithm for convolutional neural networks,” IEEE Trans. Circuits Syst. I, Reg. Papers 66 (2019) 3480 (DOI: 10.1109/TCSI.2019.2928682).

C. Yang, et al.: “A stride-based convolution decomposition method to stretch CNN acceleration algorithms for efficient and flexible hardware implementation,” IEEE Trans. Circuits Syst. I, Reg. Papers 67 (2020) 3007 (DOI: 10.1109/TCSI.2020.2985727).

J. Shen, et al.: “Toward an efficient deep pipelined template-based architecture for accelerating the entire 2-D and 3-D CNNs on FPGA,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 39 (2020) 4442 (DOI: 10.1109/TCAD.2019.2912894).

J. Jiang, et al.: “A reconfigurable Winograd CNN accelerator with nesting decomposition algorithm for computing convolution with large filters,” arXiv preprint (2021) arXiv:2102.13272.

H. Deng, et al.: “3D-VNPU: a flexible accelerator for 2D/3D CNNs on FPGA,” 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (2021) 181 (DOI: 10.1109/FCCM51124.2021.00029).

A. Lavin and S. Ko: “Fast algorithms for convolutional neural networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 4013 (DOI: 10.1109/CVPR.2016.435).

J. Yepez and S. Ko: “Stride 2 1-D, 2-D, and 3-D Winograd for convolutional neural networks,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 28 (2020) 853 (DOI: 10.1109/TVLSI.2019.2961602).

J. Yuan, et al.: “A reconfigurable Winograd-GEMM architecture,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 27 (2019) 2816 (DOI: 10.1109/TVLSI.2019.2941250).

H. Ye, et al.: “HybridDNN: a framework for high-performance hybrid DNN accelerator design and implementation,” 2020 57th ACM/IEEE Design Automation Conference (DAC) (2020) 1 (DOI: 10.1109/DAC18072.2020.9218684).

C. Yang, et al.: “WRA: a 2.2-to-6.3 TOPS highly unified dynamically reconfigurable accelerator using a novel winograd decomposition algorithm for convolutional neural networks,” IEEE Trans. Circuits Syst. I, Reg. Papers 66 (2019) 3480 (DOI: 10.1109/TCSI.2019.2928682).

C. Yang, et al.: “A stride-based convolution decomposition method to stretch CNN acceleration algorithms for efficient and flexible hardware implementation,” IEEE Trans. Circuits Syst. I, Reg. Papers 67 (2020) 3007 (DOI: 10.1109/TCSI.2020.2985727).

J. Pan and D. Chen: “Accelerate non-unit stride convolutions with Winograd algorithms,” 2021 26th Asia and South Pacific Design Automation Conference (2021) 358 (DOI: 10.1145/3394885.3431534).

J. Shen, et al.: “Toward an efficient deep pipelined template-based architecture for accelerating the entire 2-D and 3-D CNNs on FPGA,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 39 (2020) 1442 (DOI: 10.1109/TCAD.2019.2912894).

J. Jiang, et al.: “A reconfigurable Winograd CNN accelerator with nesting decomposition algorithm for computing convolution with large filters,” arXiv preprint (2021) arXiv:2102.13272.

H. Deng, et al.: “3D-VNPU: a flexible accelerator for 2D/3D CNNs on FPGA,” 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (2021) 181 (DOI: 10.1109/FCCM51124.2021.00029).

A. Lavin and S. Gray: “Fast algorithms for convolutional neural networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 4013 (DOI: 10.1109/CVPR.2016.435).

J. Yepez and S. Ko: “Stride 2 1-D, 2-D, and 3-D Winograd for convolutional neural networks,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 28 (2020) 853 (DOI: 10.1109/TVLSI.2019.2961602).

J. Wang, et al.: “FWUA: a flexible winograd-based uniform accelerator for 1D/2D/3D CNNs,” submitted to The 4th IEEE International Conference on Integrated Circuits, Technologies and Applications.

K. He, et al.: “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 770 (DOI: 10.1109/CVPR.2016.9218571).

K. Harat, et al.: “Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 6546 (DOI: 10.1109/CVPR.2018.00685).

K. Ranipa, et al.: “Multimodal CNN fusion architecture with multi-features for heart sound classification,” 2021 IEEE International Symposium on Circuits and Systems (2021) 1 (DOI: 10.1109/ISCAS51556.2021.9401551).

A. Torfi, et al.: “3D convolutional neural networks for cross audio-visual matching recognition,” IEEE Access 5 (2017) 22081 (DOI: 10.1109/ACCESS.2017.2761539).

S. Ioffe and C. Szegedy: “Batch normalization: accelerating deep neural networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 4013 (DOI: 10.1109/CVPR.2016.435).

S. Ioffe and C. Szegedy: “Batch normalization: accelerating deep neural networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 4013 (DOI: 10.1109/CVPR.2016.435).