ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

Deep Neural Networks to Accelerated by Optimize the Convolution Operation on FPGA

Girubaharan.P[1] ,Dr.K.A. Dattariya[2]

P.G Student, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India [1]

Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India [2]

ABSTRACT: As convolution contributes most operations in convolutional neural network (CNN), the convolution acceleration scheme significantly affects the efficiency and performance of a hardware CNN accelerator. Convolution involves multiply and accumulate operations with four levels of loops, which results in a large design space. Prior works either employ limited loop optimization techniques, e.g., loop unrolling, tiling, and interchange, or only tune some of the design variables after the accelerator architecture and dataflow are already fixed. Without totally learning the convolution loopoptimization before the hard section, resulting accelerator can hardly exploit thedata reuseand manage data movement efficiently. This paper overcomes these barriers by quantitatively analyzing and optimizing the look objectives (e.g., memory access) of the CNN accelerator based on multiple design variables. Then, we propose a specific dataflow of hardware CNN acceleration to minimize the data communication while maximizing the resource utilization to achieve high performance. The proposed CNN acceleration scheme and architecture are demonstrated by implementing endo-end CNNs including NiN, VGG-16, and ResNet-50/ResNet152 for inference. For VGG-16 CNN, the overall throughputs achieve 348 GOPS and 715 GOPS on Intel Stratix V and 10 FPGAs

KEYWORDS:CNN,Unrolling,Convolution Pixels, Accumulator, Tiling, Interchange

I.INTRODUCTION

THE field-programmable gate arrays (FPGA) are fast becoming the platform of choice for accelerating the inference phase of deep convolutional neural networks (CNNs). In addition to their standard blessing of reconfigurability and shorter style time over application-specific integrated circuits (ASICs) [20], [21] to catch up with speedy evolving of CNNs, FPGA can realize low latency inference with competitive energy efficiency (~10–50 GOP/s/W) Manuscript received October 27, 2017; revised February 3, 2018; accepted March 6, 2018. This work was supported in part by the NSF I/UCRC Center for Embedded Systems through NSF under Grant 1230401, Grant 1237856, Grant 1701241, Grant 1361926, Grant 1535669, Grant 1652866, and Grant 1715443; and in part by the Intel Labs, and in part by the Samsung Advanced Institute of Technology. (Corresponding author: Yufei Ma.) Y. Ma, Y. Cao, and J.-s. Seo are with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: yufeima@asu.edu; yu.cao@asu.edu; jaesun.seo@asu.edu). S. Vrudhula is with the School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: vrudhula@asu.edu). Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org. Digital Object Identifier 10.1109/TVLSI.2018.2815603

II.REVIEW OF RELATED LITERATURE

. Loop Optimization and Design Variables As shown in Fig.3, multipledimensionsare usedto describe the sizes of the feature and kernel maps of each convolution layer for a given CNN. The hardware design variables of loop unrolling and loop tiling will determine the acceleration factor and hardware footprint. All dimensions and variables used in this paper are listed in Table I. The width and height of one kernel (or filter) window is described by define the width and height of one input and output feature map (or channel), respectively denote

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

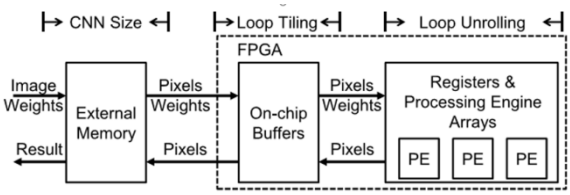
7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

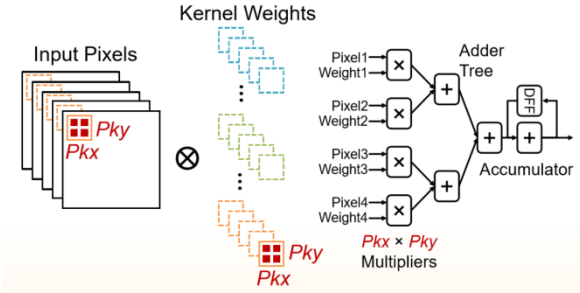
Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

the number of input and output feature maps, respectively. The loop unrolling design variables are (Pkx,Pky), Pif,(Pox, which represent the portion of data of the four loops stored in on-chip



OPTIMIZING CONVOLUTION OPERATION TO ACCELERATE DEEP NEURAL NETWORKS ON FPGA



.1) Loop Unrolling: As illustrated in Figs. 4–7, unrolling different convolution loops leads to different parallelization of computations, which affects the optimal PE architecture with respect to data reuse opportunities and memory access patterns. a) Loop-1 unrolling (Fig. 4): In this case, the inner product of pixels (or activations) and weights from different (x, y) locations in the same feature and kernel map are computed very cycle. This inner product requires an adder tree with fan-in of × to sum the parallel multiplication results, and an accumulator to add the adder tree output with the previous partial sum. Loop-2 unrolling In every cycle, number of pixels/weights from Pif different feature/kernel maps at the

Unroll loop-1and its corresponding computing architecture.

same (x, y) location are requiredto computethe inner product. The inner-product operation results in the same computing structure as in unrolling Loop-1, but with a different adder tree fan-in of c) Loop-3 unrolling (Fig. 6): In every cycle, Pix × number of pixels from different (x, y) locations in the same feature map are multiplied with the identical weight. Hence, this weight can be reused Pix × P times. Since the Pix ×

This article has been accepted for inclusion in a very future issue of this journal. Content is final as presented

III.METHODOLOGY

Loop tiling determines the size of data stored in on-chip buffers.

y parallel multiplication contributes to independent Pix × output pixels, Pix × P accumulators are used to serially accumulate the multiplier outputs and no adder tree is needed. d) Loop-4 unrolling (Fig. 7): In every cycle, one pixel is multiplied by P weights at the same (x, y) location but from P different kernel maps, and this pixel is reused P times. The computing structure is identical to unrolling Loop-3 using P multipliers and accumulators without an adder tree. The unrolling variable values of the four convolution loops collectively determine the total number of parallel MAC operations as well as the number of required multipliers (Pm)

2) Loop Tiling: On-chip memory of FPGAs is not always large enough to store the entire data of deep CNN algorithms. Therefore, it is reasonable to use denser external DRAMs to store the weights and the intermediate pixel results of all layers. Loop tiling is used to divide the entire data into multiple blocks, which can be accommodated in the on-chip buffers, as illustrated in Fig. 8. With proper assignments of the loop tiling size, the locality of data can be increased to reduce the number of DRAM accesses, which incurs long latency and high-

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

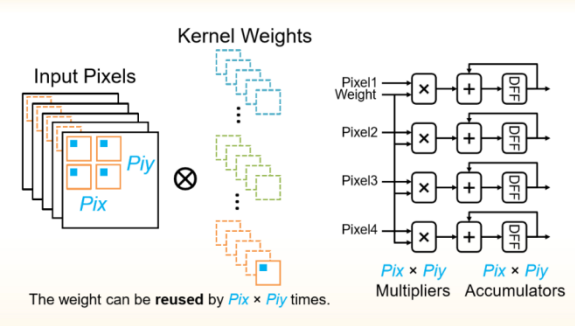
7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

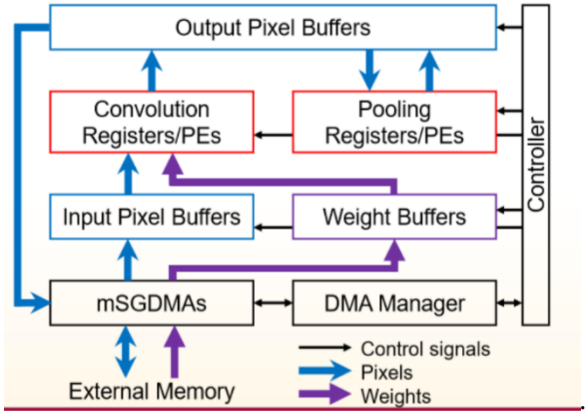
Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

power consumption. The loop tiling sets the lower bound on the required on-chip buffer size. The required size of input pixel buffer is



3) Loop Interchange: Loop interchange determines the order of the sequential computation of the four convolution loops. There are two kinds of loop interchange, namely, loop ordersloop order determines the pattern of data movements from on-chip buffer to PEs. looporderdeterminesthe data from external memory to on-chip buffer.



Data Reuse Reusing pixels and weights reduces the number of read operations of on-chip buffers. There are mainly two types of data reuse: spatial reuse and temporal reuse. Spatial reuse means that, after reading data from on-chip buffers, a single pixel or weight is used for multiple parallel multipliers within one clock cycle. On the other hand, temporal reuse means that a single pixel or weight is used for multiple consecutive cycles.
Having Pm parallel multiplications per cycle requires Pm pixels and Pm weights to be fed into the multipliers. The number of distinct weights required per cycle is Pwt =Pof ×Pif ××Pky. (9) If Loop-1 is not unrolled kx = 1, y = 1), the number of distinct pixels required per cycle (Ppx) is Ppx =Pif ×Pi. (10) Otherwise, Ppx is Ppx =Pif × ((Pix−1)x)× ((Piy−1)S+Pky). (11) Note that "distinct" only means that the pixels/weights are from different feature/kernel map locations and their values may be the same. The number of times a weight is spatially reused in one cycle is t=Pm/P (12) where the spatial reuse of weights is realized by unrolling Loop-3 (Pix > 1 or> 1). The number of times of a pixel is spatially reused in one (14) otherwise, × Pix× ((Pix−1)S+)×((Piy−1)S+ P . (15) The spatial reuse of pixels is realized by either unrolling Loop-4 (P > 1) or unrolling both Loop-1 and Loop3 together. Only unrolling Loop-1 (Pix = hampers reusing pixels, and Reuse= If intra-tile Loop-3 is computed first, the weights can be reused /(Pox ×P) consecutivecycles. If Loop-4 is computed first, the pixels can be reused for of consecutive cycles.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

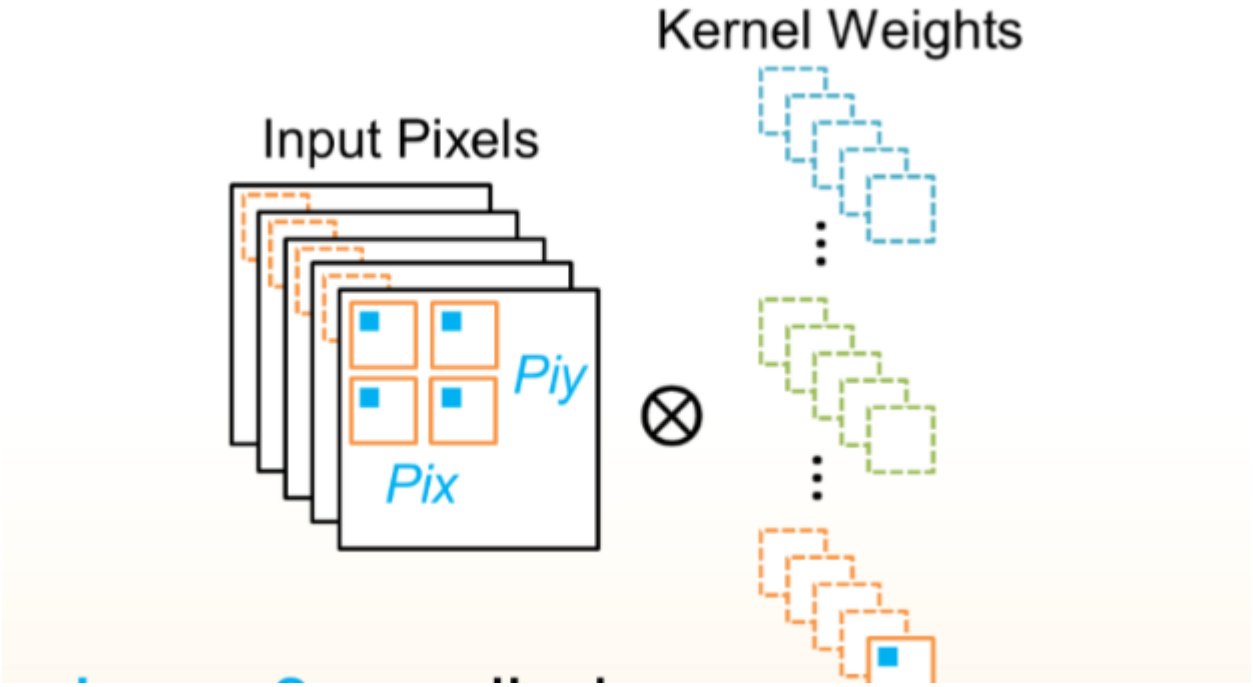
Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

D. Access of On-Chip Buffer With the data reuse, the number of on-chip buffer accesses can be significantly reduced. Without any data reuse, the total read operations from on-chip buffers for both pixels and weights are Nm, as every multiplication needs one pixel and one weight. With data reuse, the total number of read operations from on-chip buffers for weights becomes #read_=Nm/Reuse_(16) and the total number of read operations for pixels is #read_px =Nm/Reuse_px. (17) If the final output pixels cannot be obtained within one tile, their partial sums are stored in buffers. The number of write and read operations to/from buffers for partial sums per cycle is 2 ×, where all partial sums

s article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A.IMAGE ACQUISITION:



.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

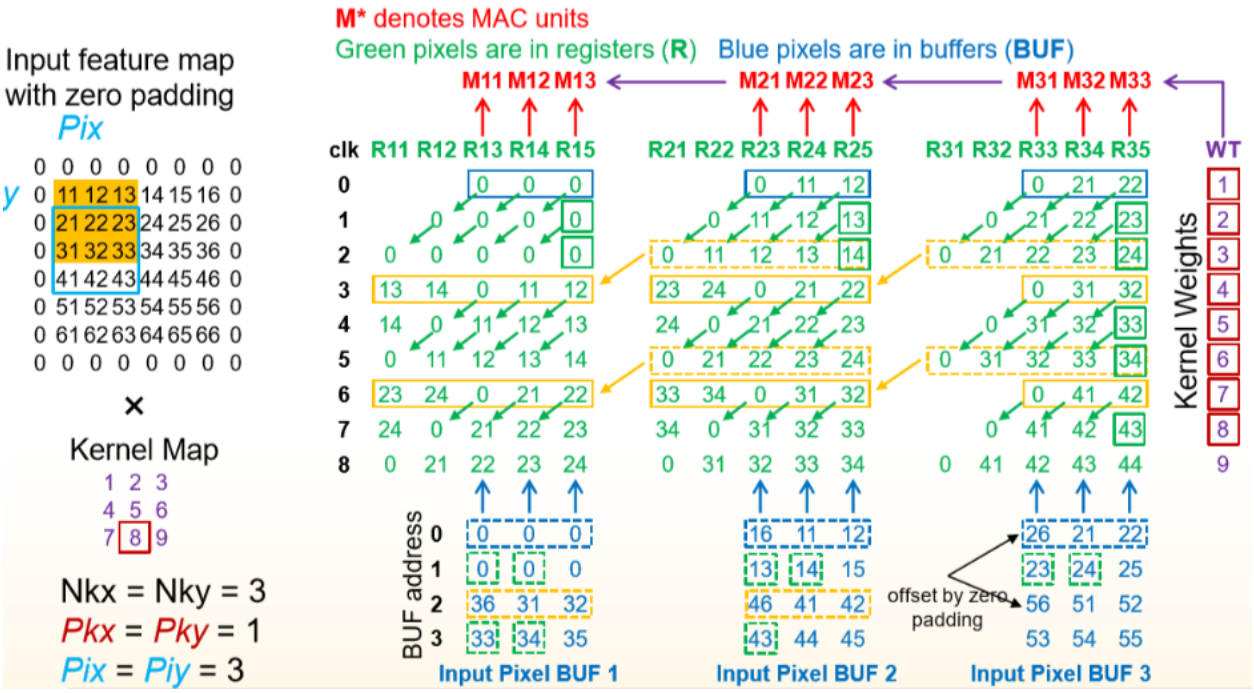
7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

B.PREPROCESSING:



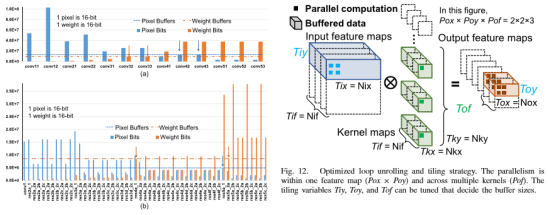


Fig. 12. Optimized loop unrolling and tiling strategy. The parallelism is within one feature map (*Pox* × *Poy*) and across multiple kernels (*Pof*). The tiling variables *Try*, *Toy*, and *Tof* can be tuned that decide the buffer sizes.

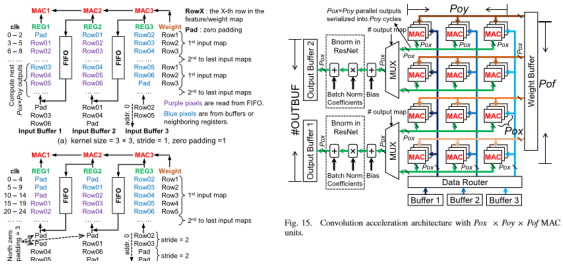


Fig. 15. Convolution acceleration architecture with *Pox* × *Poy* × *Pof* MAC units.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

IV.RESULTAND DISCUSSION

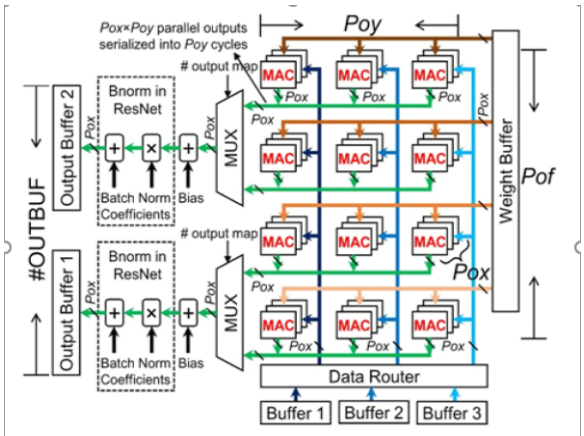
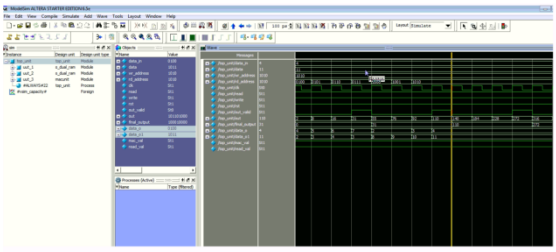


Fig. 15. Convolution acceleration architecture with *Pox* × *Poy* × *Pof* MAC units

EXPRIMENTAL OUTPUT



V.CONCLUTION

In this paper, we present an in-depth analysis of convolution loop acceleration strategy by numerically characterizing the loop optimization techniques. The relationship between accelerator objectives and design variables are quantitatively investigated. A corresponding new dataflow and architecture is proposed to minimize data communication and enhance throughput r CNN accelerator implementsend-to-end, VGG-16, and ResNet-50/ResNet-152 CNN modelse on Stratix V and 10 FPGA, achieving the overall throughput of 348 GOPS and 715 GOPS, respectively.

.

REFERENCES

[1] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.
[3] M. Lin, Q. Chen, and S. Yan. (Mar. 2014). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400
[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[6] D. F. Bacon, S. L. Graham, and O. J. Sharp, "Compiler transformations for high-performance computing," ACM Comput. Surv., vol. 26, no. 4, pp. 345–420, Dec. 1994
[7] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energyefficient reconfigurable accelerator for deep convolutional neural networks," IEEE J. Solid-State Circuits, vol. 51, no. 1, pp. 127–138, Jan. 2017.
[8] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in Proc. ACM/IEEE Int. Symp. Comput. Archit. (ISCA), Jun. 2016, pp. 367–379.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 8, Special Issue 1, March 2019

7th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '19)

19th-20th March 2019

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

[9] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays (FPGA), Feb. 2015, pp. 161–170.
[10] C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong, "Energy-efficient CNN implementation on a deeply pipelined FPGA cluster," in Proc. ACM Int. Symp. Low Power Electron. Design (ISLPED), Aug. 2016, pp. 326–331.
[11] N. Suda et al., "Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks," in Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays (FPGA), Feb. 2016, pp. 16–25.
[12] U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An OpenCL deep learning accelerator on Arria 10," in Proc. ACM/SIGDA Symp. Field-Program. Gate Arrays (FPGA), Feb. 2017, pp. 55–64.
[13] K. Guo et al., "Angel-Eye: A complete design flow for mapping CNN onto embedded FPGA,"IEEETrans. Comput.-Aided Des. Integr. Circuits Syst., vol. 37, no. 1, pp. 35–47, Jan. 2018.