Check for updates

A Conv-GEMM reconfigurable accelerator with WS-RS dataflow for high throughput processing

Feihu Wang,[1,2] Chi Zhang,[1,2] Yongchao Deng,[1,2] Xu Yang,[1] Shuangming Yu,[1,2] Runjiang Dou,[1,2] Nanjian Wu,[1,2] and Liyuan Liu[1,2,✉]

[1] *State Key Laboratory of Superlattices and Microstructures, Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China*

[2] *University of Chinese Academy of Sciences, Beijing, China*

✉ E-mail: liuly@semi.ac.cn

Convolution and matrix operations are both important computations in Deep Neural Networks (DNNs). However, the significant differences between convolution and matrix computation patterns have posed a challenge in efficiently supporting both convolution (Conv) and general matrix multiplication (GEMM) on hardware design. This paper proposes a Conv-GEMM reconfigurable accelerator architecture for high throughput edge processing. A weight stationary-row streaming (WS-RS) dataflow scheme is proposed, which maximizes data reuse through hierarchical memory structures and flexible PE connections, and supports high throughput edge-based deep learning algorithms. Based on the proposed dataflow, multi-scale memory access network (MMAN), reconfigurable accumulator array (RAA), and configurable instruction set architecture (ISA) are designed to optimize computation throughput and energy efficiency. The accelerator is designed under 65 nm technology, achieves peak performance of 1.15 TOPS at 250 MHz, with an energy efficiency of 1.14 TOPS/W. The GEMM computation achieves 85.7% latency improvement and the Mobilenet-V1 processing achieves a throughput of 529 fps under a 256 × 224 image size and an 87.15% (top-5) accuracy on the ImageNet dataset.

Introduction: Edge neural network accelerators prioritize reducing the latency and power consumption of Deep Neural Networks (DNNs) inference, which requires optimization of convolution (Conv) in convolution neural networks (CNNs) and general matrix multiplication (GEMM) in the transformer's attention block. A unified architecture for Conv and GEMM can improve overall throughput of networks mentioned above. The GEMM implementation for Conv combines two types of operations, and enhances computational resource efficiency by addressing the I/O performance drawbacks of irregular Conv. However, it incurs redundant storage space and is not ideal for edge devices. Due to the significant differences in computation patterns between Conv and GEMM, simultaneously improving the throughput and energy efficiency of both operations has become a challenge.

As Figure 1 shows, current accelerators are divided into 2D mapping and systolic architectures. 2D mapping accelerators [1] implement fine-grained operations through SIMD, which provides strong programmability, but their throughput is limited by memory bandwidth and complex broadcast networks. Systolic architecture accelerators [2–4] achieve high throughput but has lower flexibility. Moreover, they optimize either Conv or GEMM.

Conv and GEMM are highly suitable for systolic architectures due to their regular dataflow. The [3] is a semi-systolic Conv accelerator with a complex multi-cast network due to weight sharing between row PEs, resulting in higher bandwidth and redundant storage space. Reference [6] reuses GEMM hardware for Conv processing. However, the ifmap vector is shared by the horizontal lines across all PE array columns through broadcasting, resulting in long critical path. Moreover, the on-the-fly

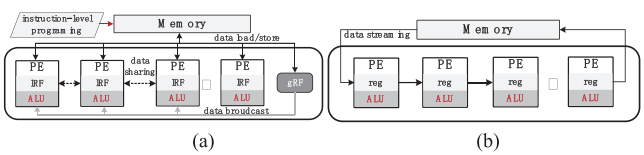


Fig. 1 *(a) 2D Mapping architecture (lRF and gRF is local and global regfile, respectively). (b) Systolic architecture*

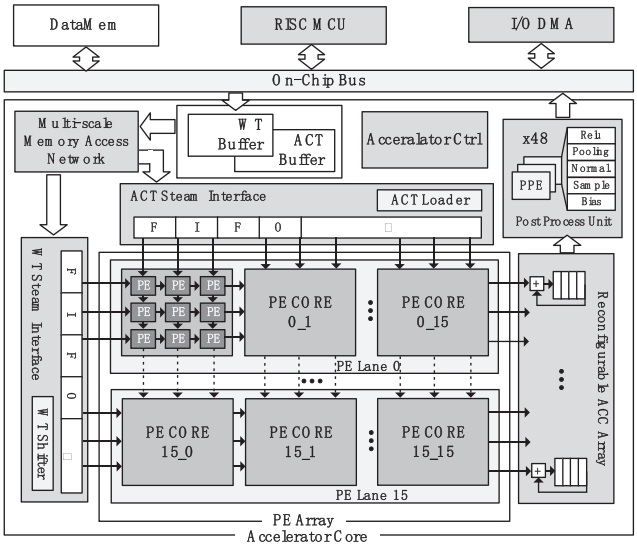


Fig. 2 *Overall architecture*

Im2Col hardware for mapping depthwise Conv to GEMM requires line buffer and control logic for Conv lowering. Reference [2] implements a large full-systolic array to support GEMM, achieving lower bandwidth requirements and simple PE structure. However, it requires additional or matrix rotation module [7] to rearrange data for Conv, incurs additional storage and logic overhead, resulting in significant latency.

To address the above problems, we propose a reconfigurable and energy-efficient accelerator for high throughput edge processing, which efficiently supports both Conv and GEMM by optimizing dataflow. A WS-RS dataflow that effectively reduces storage and bandwidth requirements, improves throughput by exploiting data reuse is proposed. To support WS-RS dataflow, improve hardware utilization and throughput, multi-scale memory access network (MMAN) and reconfigurable accumulator array (RAA) is proposed.

Architecture overview: Figure 2 shows the overall architecture of the proposed accelerator. It consists of a reconfigurable accelerator core (AC), an IO DMA, a DataMem (DM), and a RISC MCU. The top RISC MCU manages the data transmission path of the I/O DMA and monitors task status of the AC. If the on-chip memory can hold all the parameters of a network, there is no external access to DRAM. Otherwise, the MCU will control the high-speed I/O DMA to pre-fetch the intermediate weights in DRAM, hiding the additional access latency. AC includes a 48 × 48 PE array (PA), two global buffers (ACT Buffer, WT Buffer), two data streaming interfaces (ACT Stream Interface, WT Stream Interface), a RAA, a MMAN, a post processing unit (PPU), and an accelerator controller (ACtrl). The PA is divided into 16 PE lanes (PL), a PL comprises 16 PE cores (PC), which accumulates the input channels of the Conv layer and generates the result for one output channel. A PC comprises 3 × 3 PEs, which performs a 3 × 3 convolution. The 1MB ACT buffer and 4KB WT buffer stores 8-bit input and weight, respectively. The act stream interface (ASI) consists of an ACT loader and an ACT FIFO, for arranging and streaming input into the PA. The WT stream interface (WSI) includes a WT shifter and a WT FIFO, for shifting weight into the PA. The RAA accumulates inter-tile partial sums (psum). The PPU performs operations such as normalization and activation. The ACtrl decodes configuration instructions and updates the control flows of AC.

WS-RS dataflow and data reuse: AC has three levels of the memory hierarchy: local registers in PE, regfiles (FIFO) in stream interfaces, and global buffers. AC reduces memory access power consumption by minimizing access to high-cost buffer, and maximizing reuse of data stored in registers and regfiles located close to PEs. Figure 3a shows the proposed WS-RS dataflow. The weights for Conv are fixed in PA using weight stationary (refer to WS), while the input feature maps are streamed into PA in a row streaming (refer to RS) fashion following a certain time sequence. Three weights are fixed in three PEs. The psums and the inputs

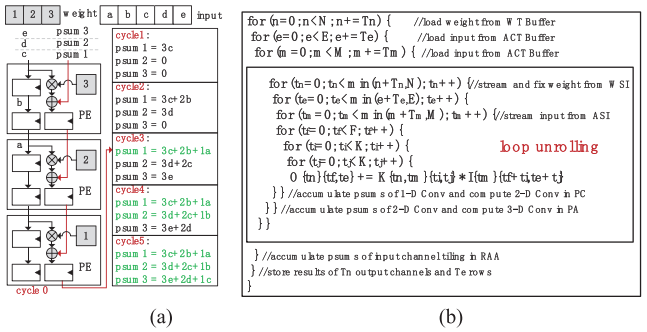


Fig. 3 *(a) The WS-RS dataflow computing 1D Conv. (b) Nested loop of Conv computation in PA*

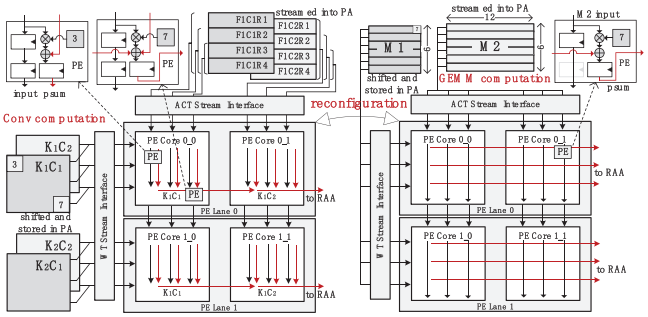


Fig. 4 *Conv (left) and Blocked GEMM (right) Reconfiguration. F denotes feature map, C denotes channel, R denotes row, K denotes kernel*

are propagated downward. Input is registered in PE for two cycles, and three 1D Conv results are obtained in five cycles.

AC has good scalability and can meet large Conv and GEMM requirements by increasing the size of PA. If the size of Conv or GEMM exceeds the size of PA, as nested loop shown in Figure 3b, the task is divided, and inter-tile psums are accumulated in RAA. Consider a Conv layer of M input channels, N output channels, kernel size of $K \times K$, input feature map size of $H \times W$, and output feature map size of $E \times F$. Tiling for input channel and output channel are T_m and T_n, adapting to the width and height of PA. Tiling for output rows is T_e, adapting to the size of RAA. WS-RS dataflow maximizes reuse of input by streaming input across T_n output channels, each input pixel is reused $K \times K$ times within a filter plane, resulting a total reuse of $T_n \times K \times K$ times. Weight is reused $T_e \times F$ times within a filter plane. Psums of different input channel are accumulated across T_m input channels, each 1D Conv psum is reused K×K times, resulting a total reuse of $T_m \times K \times K$ times. Input, weight and psum of one tiling are pipelined and shared between low-cost registers in PEs, reducing power consumption.

Reconfiguration of Conv and GEMM: Figure 4 shows an example of how AC is reconfigured to compute Conv and GEMM. Two input channels and two output channels of weights are fixed in 4 PCs. Two input channels of input feature map stream into 2 PCs sequentially delayed by 1 cycle, the 1D psums and inputs are propagated downward within each PE column. The 2D psums are further collected and accumulated towards the right at the bottom of each PC using 3-input adder. The 3D psums of different input channels are accumulated through column-wise PC connections in a PL, and inputs are reused across output channel through row-wise PL connections in PA. The mapping of GEMM uses block GEMM approach as the same Weight-Stationary dataflow as [2] by modifying the timing path within PE and connections between PC. Matrix M1 is fixed in PA, while matrix M2 streams into column PEs and psums are accumulated through row PEs. PA also supports kernel size of 1×1 and 5×5 by dividing a PC into nine independent PEs or grouping 4 PCs as a large PC.

PE structure: Each PE consists of a MAC unit, a weight path register (path_reg), a weight register (wt_reg), an input register (act_reg), a partial sum register (psum_reg), a wait register (wait_reg), three data selectors (mux1, mux2, mux3), a MAC and PE Ctrl unit, as shown in

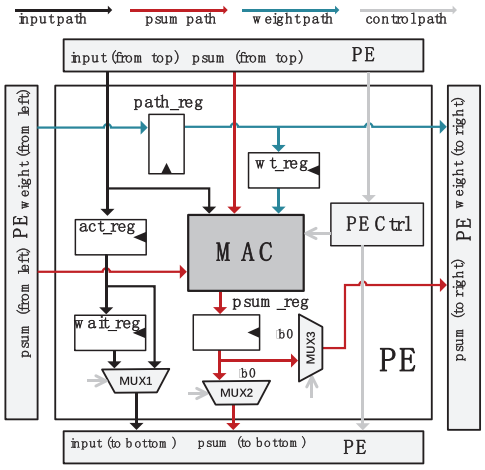


Fig. 5 *Schematic of PE*

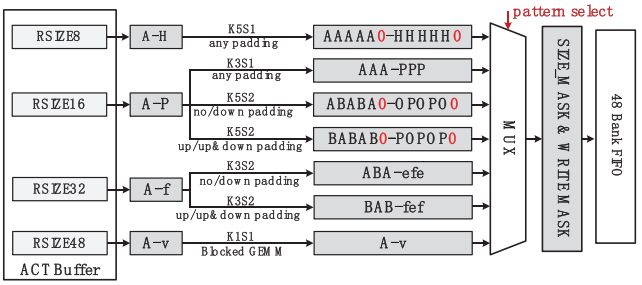


Fig. 6 *The multi-scale memory access network for Conv and GEMM*

Figure 5. PE has a horizontal weight path, a vertical input path, and two psum paths, horizontal and vertical direction respectively. As Figure 5 shows, mux1 selects wait_reg's value when computing Conv layer, otherwise selects the value of act_reg when calculating GEMM. When computing Conv layer, psum propagates from top to bottom across first two rows' PEs in PC, and propagates from left to right across PEs in the third row of PC by controlling mux2 and mux3. The path_reg stores weight shifted to PE, overlaps weight loading and Conv computing to reduce latency. This simple PE structure supports computation of various Conv and GEMM data streams with minimal hardware overhead. By using simple and uniform PEs, the global broadcast network and fan in are eliminated, consuming limited bandwidth and power consumption to achieve high throughput.

Multi-scale memory access network: MMAN can handle memory access inconsistency while leverage the data reuse achieved by the WS-RS dataflow to further improve energy efficiency and provide enough bandwidth to support the highly parallel processing in the PE array.

Figure 6 shows The MMAN for Conv and GEMM. The data fetched from memory undergoes replication and rearrangement according to computation type. Pattern select signals choose one of the data paths, which is then written to the ASI's FIFO after selection based on the size mask for input and output channel numbers and the write mask for padding strategies. The loading network for weights is similar to that of input's.

Reconfigurable accumulator array: RAA has a 128-bit 48-bank dual port accram located to the right of PA and is used to receive psums between Conv or GEMM tilings. The psums generated by PA are passed out in a diagonal fashion. One read operation and one write operation happens concurrently in accram, fully utilizing the dual-port SRAM with its two ports. Inter-tiling computation is overlapped, which reduces the latency of network inferences.

Depthwise Conv does not require psum accumulation across input channels, and requiring larger bandwidth compared to regular Conv. RAA can be reconstructed into Input mode, serving as a data rearrangement buffer, as Figure 7 shows. Every three banks of accram are grouped

1350911x, 2024, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.13125 by Institute of Semiconductors, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

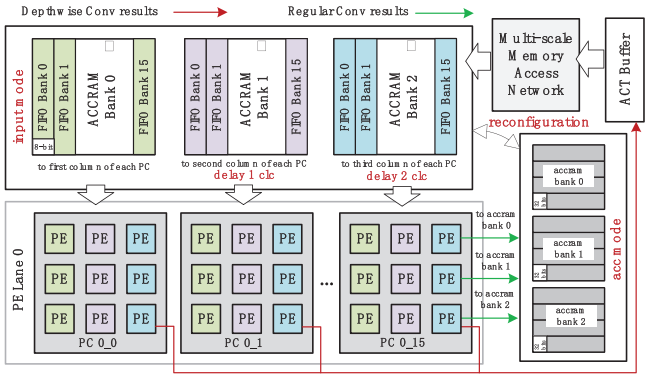
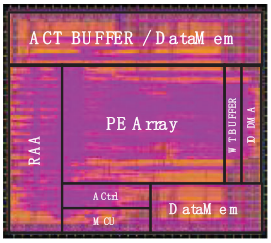


Fig. 7 *The reconfiguration of input mode and acc mode of reconfigurable accumulator array*



Technology	65 nm
Die Area	9mm x 9mm
Voltage	1.2 V
Power	1.01 W
SRAM	5.5 MB
Max Frequency	250 MHz
Peak Performance	1.15 TOPS

Fig. 8 *Chip layout and specifications*

as FIFO between ACT Buffer and a PL, and each 128-bit accram bank is divided into 16 small 8-bit FIFO banks, feeding data into the 16 PCs in a PL. The input streaming is delayed by one cycle between accram banks. The reconfiguration of RAA increases the throughput of depthwise Conv by ×256.

Chip implementation: We implement the accelerator in 65 nm technology using EDA tools. Figure 8 shows chip layout photograph and chip specifications. The accelerator works at a maximum frequency of 250 MHz with 1.2 V, consuming 1.01 W reported by PrimeTime.

Evaluation: The accelerator adopts WS-RS dataflow, which maximizes reuse of inputs, weights and psums of Conv, efficiently combines Conv and GEMM. MMAN and RAA are proposed to improve bandwidth utilization and computation throughput. We compare the cycle number required for MobileNet-V1 between the accelerator's WS-RS dataflow and [2]'s im2col and rotate matrix process. The im2col process converts 3D convolution into GEMM by replicating and arranging the input, requiring each input to be copied $K \times K$ times (including padding), which leads to redundant storage space, shown in Figure 9a. The rotation matrix algorithm transforms the input into a specific vector

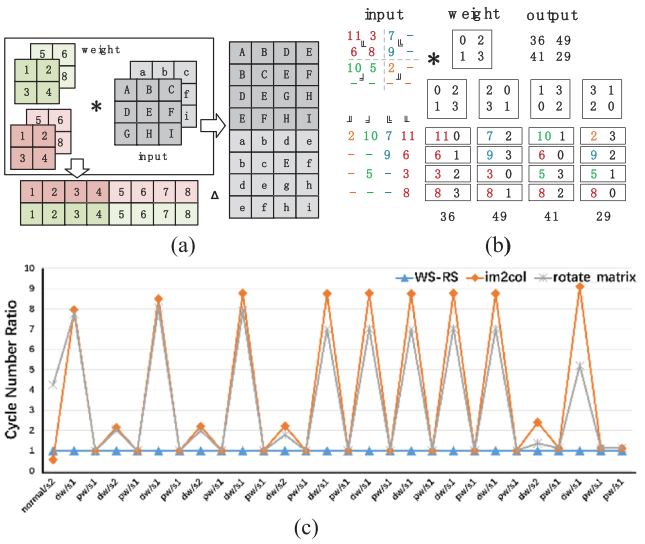


Fig. 9 *(a) The im2col process of Conv. (b) The rotate matrix process of Conv. (c) The cycle ratio of different methods computing MobileNet-V1, dw denotes depthwise Conv, pw denotes pointwise Conv, s1 denotes stride 1*

format, serving as row inputs for the PE array. The 2 × 2 kernel is rotated, and split into four matrices that are column input for the PE array, shown in Figure 9b. As the result shown in Figure 9c, WS-RS dataflow requires the minimum number of cycles with no redundant storage for the template 9 × 9 PE array. The im2col process consumes more clock cycles computing depthwise Conv because the vanilla TPU-like array is not optimized for single-channel Conv. We also compared GEMM between the accelerator and a typical 2D-Mapping processor [1], which supports SIMD fashion vector computation. When performing a GEMM of a 16 × 1024 matrix and a 1024 × 1024 matrix, our accelerator achieves 85.7% latency improvement.

Moreover, we deploy an image classification and object tracking task on the accelerator. The image classification task uses the 24-layer MobileNet-V1 for ImageNet dataset, applying to images of 256 × 224 × 3 size. The processing latency is 1.89 ms per frame with an accuracy of 87.15% (top-5). The object tracking task uses the 16-layer SiamFC-Mobile for OTB100 dataset, tracking a 128 × 128 × 3 target from a 256 × 256 × 3 background. The processing latency is 2.15 ms per frame.

Comparison with previous works: Table 1 presents the comparisons of our accelerator and previous state-of-the-art DNN processors. Reference [2] can perform Conv and GEMM, but it uses im2col or rotate matrix process to compute Conv, and lacks optimization for depthwise Conv. References [3–5] only support Conv computation. Reference [3] adopts row-stationary dataflow to compute Conv, but it has complex

Table 1. Comparison with previous works

Metrics	ISCA'17 [2]	JETCAS'19 [3]	ISSCC'21 [5]	ISSCC'22 [4]	Ours
Dataflow	WS, fully-systolic	RS, semi-systolic	SIMD, 2d-mapping	WS, fully-systolic	WS-RS, fully-systolic
Technology	28 nm	65 nm	22 nm	65 nm	65 nm
Area (mm^2)	331	2695 gates	62	4.47	81
Voltage (V)	0.8–1.0	0.82–1.17	0.8	1	1.2
Freq. (MHz)	700	200	262.5	400	250
SRAM	28 MB	246 kB	9 MB	180 kB	5.5 MB
Bit-width	8	8	8/16/32	8	8
Operation	Conv, Matrix Op.	Conv	Conv	Conv	Conv, Matrix Op.
Energy efficiency	0.84* TOPS/W	0.415 TOPS/W	1.065* TOPS/W	0.66 TOPS/W	1.14 TOPS/W
Throughput	-	120fps@ AlexNet	322fps@ MobileNet-V1	51fps@ ResNet18	529fps@ MobileNet-V1

*Power scaled according to [8].

1350911x, 2024, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.13125 by Institute of Semiconductors, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

PE structure and NoC, which limits its array size. Reference [5] uses 2D-mapping architecture, resulting in relatively low energy efficiency. Reference [4] does not fully exploit data reusability. The scaled power method is reported in [8].

Conclusion: This brief proposes an accelerator that supports various Conv and matrix operations. It utilizes a WS-RS dataflow, effectively reducing additional storage requirements, input rearrangement logic, and minimizing data transmission latency. It maximizes the reuse of various data types and employs psum accumulation paths, eliminating the need for a global broadcast network and addition trees, thereby reducing hardware overhead and improving energy efficiency.

Author contributions: **Feihu Wang**: Conceptualization; investigation; methodology; writing—original draft; writing—review and editing. **Chi Zhang**: Validation. **Yongchao Deng**: Validation. **Xu Yang**: Project administration. **Shuangming Yu**: Project administration. **Runjiang Dou**: Project administration. **Nanjian Wu**: Supervision. **Liyuan Liu**: Supervision.

Acknowledgements: This research was funded in part by the National Key Research and Development Program of China under Grant 2019YFB2204303; in part by the National Natural Science Foundation of China under Grant U20A20205, U21A20504 and 62274154; in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB0460000; in part by the Beijing Natural Science Foundation Z220005, Beijing Municipal Science and Technology Project Z221100007722028.

Conflict of interest statement: The authors declare no conflicts of interest.

Data availability statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

© 2024 The Authors. *Electronics Letters* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
Received: *11 December 2023* Accepted: *1 February 2024*
doi: 10.1049/ell2.13125

References

1 Luo, Q., et al.: A programmable and flexible vision processor. *IEEE Trans. Circuits Syst. II* **69**(9), 3884–3888 (2022)
2 Jouppi, N.P. et al.: In-datacenter performance analysis of a tensor processing unit. In: *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture*. Toronto, Canada, pp. 1–12 (2017)
3 Chen, Y.-H., et al.: Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE J. Emerging Sel. Top. Circuits Syst.* **9**(2), 292–308 (2019)
4 Ju, Y., et al.: A 65 nm systolic neural CPU processor for combined deep learning and general-purpose computing with 95% PE utilization, high data locality and enhanced end-to-end performance. In: *2022 IEEE International Solid-State Circuits Conference*. San Francisco, United States, pp. 1–3 (2022)
5 Eki, R., et al.: 9.6 a 1/2.3 Inch 12.3 Mpixel with on-chip 4.97 TOPS/W CNN processor back-illuminated stacked CMOS image sensor. In: *2021 IEEE International Solid-State Circuits Conference*. San Francisco, United States, pp. 154–156 (2021)
6 Manasi, S.D., et al.: Reusing GEMM hardware for efficient execution of depthwise separable convolution on ASIC-based DNN accelerators. In: 2023 28th Asia and South Pacific Design Automation Conference (ASP-DAC). Tokyo, Japan, pp. 475–482 (2023)
7 Ross, J., et al.: Rotating data for neural network computations. *US Patent*, US20160342893A1 (2016)
8 Stillmaker, A., et al.: Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. *Integration* **58**, 74–81 (2017)

1350911x, 2024, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.13125 by Institute of Semiconductors, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License