Dean&Francis
ISSN 2959-6157

Low-Power Hardware Implementation and Evaluation of Sparse CNN for MNIST

Junze Ren*

School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, 100876, China
*Corresponding author: 2022211915@bupt.cn

Abstract:

To address the low-power requirements of edge computing, this work takes MNIST classification as a case study and investigates a sparsity-driven lightweight CNN hardware implementation. Under a unified toolchain, we evaluate the gate-level power of the first convolution–pooling block (Conv1) using VCD-based switching-activity analysis. A sparsity sweep is conducted to study the energy–accuracy trade-offs of zero-skipping. On a 500-image evaluation subset, the baseline achieves 98.0% accuracy with 0.236 μW power. Increasing sparsity to 65% reduces Conv1 power to 0.164 μW (30.5% lower) with 96.0% accuracy. When accuracy must remain at the 98.0% baseline level, a 45% sparsity setting yields 0.182 μW power. The comparison between Zero-Skip and Zero-Skip+CE further indicates that the current CE implementation behaves as data gating rather than true clock gating, providing no additional energy benefit.. We provide detailed descriptions of model configuration, sparsity and gating implementation, VCD statistics and power-conversion settings required for reproducible experiments, thereby offering quantitative evidence and engineering guidelines for sparsity selection and energy-efficiency optimization of lightweight CNNs on resource-constrained platforms.

Keywords: Sparse CNN, Zero-skip, Clock enable, VCD-based power estimation, Low-power accelerator.

1. Introduction

As energy-efficiency requirements in edge and embedded scenarios keep increasing, deploying lightweight CNNs on low-compute platforms must achieve a controllable trade-off between accuracy and power consumption. Sparsity and gating are regarded as effective means to reduce switching activity and memory access cost, and many works have systematically surveyed pruning and compression methods and their impact on performance and energy efficiency [1,2]. At the hardware level, existing studies report significant energy-efficiency improvements from pruning-induced sparsity, data and clock gating, computation--memory co-design and model quantization. For example, EIE realizes highly efficient inference on compressed sparse models [3,4]; SCNN and NullHop exploit both weight and activation sparsity,

ISSN 2959-6157

using compressed storage and zero-skipping to reduce MAC operations and memory traffic [5,6]; Eyeriss leverages spatial dataflow and data reuse to lower on-chip and off-chip memory energy [7]; and on low-density FPGAs, zero-skipping and weight pruning have been integrated into unified CNN accelerators to achieve high throughput and energy efficiency for edge scenarios [3]. Integer quantization further improves latency and energy efficiency while maintaining accuracy close to floating-point inference [8]. However, these works mostly focus on throughput, bandwidth, and overall energy-efficiency metrics. Although VCD-based power estimation is widely used in EDA flows, its standardized use and reproducibility in course-level or teaching-level prototypes remain limited, and there is a lack of reproducible evidence under a unified methodology regarding the true accuracy--power relationship of the "zero-skipping + CE" combination.

To this end, we build an evaluation flow under a unified experimental framework that consists of "sparsity sweep--VCD counting--power conversion--automatic plotting". CE is explicitly defined as a write-enable gating signal generated by consecutive-zero detection in front of the MAC (only blocking register writes while the clock network still toggles). The flow reveals the trade-offs between sparsity and energy-efficiency under zero-skipping, provides the minimum-power choice under accuracy constraints, and, through a "Zero-Skip vs Zero-Skip+CE" comparison, shows that without turning off the clock network, no additional power benefit is obtained, suggesting that real clock gating is required. The paper is organized as follows: first, we present the model and environment, sparsity and gating implementation, and VCD-based estimation method; then we report experimental results and deployment recommendations; finally, we discuss conclusions and directions for improvement.

2. Methods

This study uses MNIST handwritten digit classification as the benchmark task and adopts a lightweight convolutional network structure (Conv1→Pool1→Conv2→Pool2→FC) [9]. The train/validation/test data split is kept consistent for both training and inference, and random seeds and software dependencies are fixed to ensure reproducibility. The network is quantized to INT8 and trained using Adam (learning rate 0.001) for 10 epochs, achieving about 99% validation accuracy. The quantization configuration follows common integer-arithmetic-only inference practices, which provide latency and energy improvements on edge devices while maintaining accuracy close to floating point [8].

On the full test set, we sequentially select the first 500 images to form an evaluation subset. The unpruned baseline achieves 98.0% accuracy on this subset. All accuracy figures reported in this paper, such as 98.0%, 96.0%, 79.0%, and 48.0%, are computed on this 500-image evaluation subset, with the goal of characterizing the relative accuracy changes across different sparsity and gating configurations rather than pursuing state-of-the-art absolute performance on MNIST.

Sparsity is introduced by magnitude-based pruning of weights with a specified sweep range and step size, so that the MAC units can skip unnecessary computations whenever zero elements are encountered. The hardware path implements skipping via zero detection and data-selection logic, ensuring functional equivalence and controllable latency. Specifically, we adopt a magnitude-based pruning strategy that preserves weights with the largest absolute values. The sparsity level is swept from 0% to 80% in 5% steps, and a bitmap index is used to support single-cycle zero detection and MAC-unit gating. As sparsity increases, the proportion of invalid operations is gradually reduced. This design is consistent with sparse CNN accelerators that exploit zero weights and activations to skip redundant MAC operations [3][5][6].

In this work, CE is explicitly defined with data-gating semantics. CE is generated before the MAC by a consecutive-zero detector and is used as a write-enable signal: registers are only updated when data need to be written. When consecutive zeros indicate that the current cycle is invalid, CE disables register writes to reduce switching activities on the data path, while the clock network and the clock pins of flip-flops still toggle. Therefore, the achievable power savings heavily depend on the proportion and continuity of zeros, and the upper bound is lower than that of true clock gating (e.g., integrated clock gating cells inserted by synthesis tools, ICG).

To relate switching activity to power consumption, we perform gate-level simulations on the above 500-image evaluation subset. For each sparsity and gating configuration, 100 images are selected for gate-level simulation to generate a full VCD file (about 900~MB per run). We then separately count toggles on data-path signals and clock signals. Power conversion follows the classical dynamic power model:

$$P_{\text{dyn}} = \alpha C_{\text{eff}} V^2 f \tag{1}$$

where α is the activity factor obtained from VCD statistics, C_{eff} is the effective load capacitance, V is the supply voltage, and f is the clock frequency [10]. To obtain normalized comparisons under a consistent methodology, the measured toggles of the unpruned baseline configuration on Conv1 are mapped to a baseline power $P_{\text{base}} = 0.236$

μW. The power of other configurations is scaled proportionally according to the ratio of their activity factor to that of the baseline:

$$P = P_{\text{base}} \cdot \frac{\alpha}{\alpha_{\text{base}}} \tag{2}$$

so that relative power variations introduced by different sparsity and gating options can be captured without relying on physical dimensions and process details. All schemes are evaluated relative to this baseline to guarantee consistent interpretation in figures and tables.

Experiments are run on a unified platform and toolchain to avoid cross-platform bias. The software environment includes PyTorch 1.12+ for training and evaluation, ModelSim SE/DE 10.5+ for gate-level simulation, and Python 3.9 with matplotlib, pandas, and openpyxl. The hardware platform is a single fixed workstation with unified CPU/GPU, memory, and operating system. An automation script auto_experiment.py completes the full pipeline of "sparsity setting → RTL configuration and compilation → simulation (100 images per configuration) → VCD power analysis → accuracy computation → report and plot generation". In total, 33 configurations are run (17 sparsity levels × 2 modes plus 1 reference), and scripts, parameters, and version information are automatically logged. The script outputs structured CSV/Excel reports and two 300~DPI PNG figures (accuracy vs. sparsity and power vs. sparsity).

3. Results

3.1 Impact of sparsity on accuracy and power

Figure 1 shows the impact of sparsity on accuracy and power under the Zero-Skip mode. Accuracy is measured on the evaluation subset consisting of the first 500 images of the MNIST test set, and the power results are obtained by performing gate-level simulations on 100 images per configuration within this subset, counting VCD toggles and converting them into power values according to the unified model. Overall, the results are derived from evaluations on the first 500 MNIST test images using VCD-based power estimation.

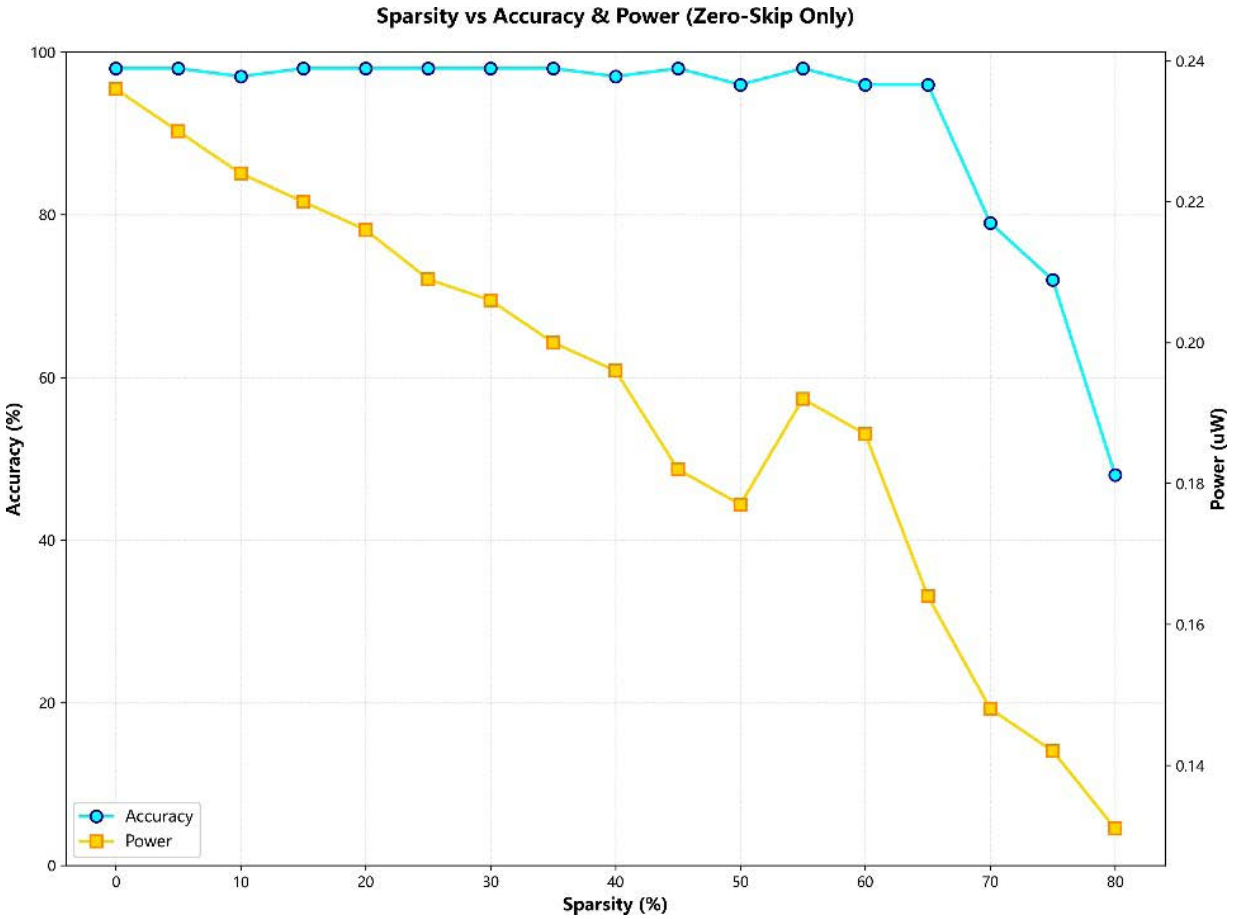

Fig. 1 Impact of sparsity on accuracy and power. The left y-axis is accuracy and the right y-axis is Conv1 power.

ISSN 2959-6157

(Picture credit : Original)

For the unpruned baseline, the model achieves 98.0% accuracy on the 500-image subset, and Conv1 power is 0.236μW. As sparsity increases from 0% to 65%, Conv1 power monotonically decreases to 0.164μW, corresponding to a 30.5% reduction from the baseline, while accuracy is 96.0%. This demonstrates a clear power--accuracy trade-off.

The power reduction mainly comes from the decrease in switching activity on the data path. In the baseline configuration, there are about 14.1M data-path toggles, whereas at 65% sparsity the number drops to about 4.7M (a 66.7% reduction). The proportion of clock toggles across configurations remains in the range of 13%--15%. If we require accuracy on the 500-image subset not to be lower than the baseline 98.0%, then 45% sparsity can be selected, reducing Conv1 power to 0.182μW (a 22.9% reduction) while maintaining 98.0% accuracy. Under a relaxed constraint that allows accuracy to drop by at most 2 percentage points relative to the baseline, 65% sparsity becomes a more aggressive choice, yielding a 30.5% power reduction with 96.0% accuracy.

However, when sparsity exceeds 65%, accuracy degrades sharply. On the same evaluation subset, 70% sparsity leads to an accuracy of 79.0% (a drop of 17 percentage points), and at 80% sparsity accuracy further decreases to 48.0%. This indicates that over-pruning severely damages the feature-extraction capability of Conv1, such that later layers can no longer compensate for the lost information.

Based on all sparsity levels and both modes, Table 1 summarizes Conv1 accuracy, power, and power reduction relative to the baseline on the 500-image evaluation subset.

Table 1. Accuracy and power of Conv1 on the first 500 test images under different sparsity levels and modes

Sparsity (%)	Zero-Skip			Zero-Skip+CE		
	Acc. (%)	Power (μW)	Red. (%)	Acc. (%)	Power (μW)	Red. (%)
0	98	0.236	0			
5	98	0.23	2.5	98	0.23	2.5
10	97	0.224	5.1	97	0.224	5.1
15	98	0.22	6.8	98	0.22	6.8
20	98	0.216	8.5	98	0.216	8.5
25	98	0.209	11.4	98	0.211	10.6
30	98	0.206	12.7	98	0.207	12.3
35	98	0.2	15.3	98	0.202	14.4
40	97	0.196	16.9	97	0.199	15.7
45	98	0.182	22.9	98	0.185	21.6
50	96	0.177	25	96	0.179	24.2
55	98	0.192	18.6	98	0.196	16.9
60	96	0.187	20.8	96	0.192	18.6
65	96	0.164	30.5	96	0.171	27.5
70	79	0.148	37.3	79	0.154	34.7
75	72	0.142	39.8	72	0.148	37.3
80	48	0.131	44.5	48	0.139	41.1

3.2 Impact of CE on energy efficiency

Figure 2 shows the power--accuracy relationship of the baseline, Zero-Skip, and Zero-Skip+CE modes. It can be observed that, at the same sparsity level, enabling CE does not reduce power; instead, Conv1 power is slightly higher than with CE disabled. At low sparsity levels (5%--25%), the power increase is about 0.3%--1.0%, while at high sparsity levels (50%--80%), the increase is about 1.1%--6.2%. For example, at 65% sparsity, power is 0.164μW without CE, whereas enabling CE increases power to 0.171μW (an increase of 0.007μW, or about +4.3%). Results are obtained from evaluations on the first 500 images of the MNIST test set using VCD-based Conv1 power estimation.

This counter-intuitive result is consistent with the implementation characteristics where CE acts as data gating while the clock network continues to toggle. Although CE suppresses register writes on zero activations, the ad-

ditional gating-control logic (consecutive-zero detector, gated_cycles counters, etc.) introduces extra switching overhead, and the clock-tree power is not reduced. This indicates that, under the current definition, CE does not provide noticeable energy savings in the clock domain. In terms of accuracy, enabling or disabling CE yields identical results at the same sparsity level, confirming that CE only affects power consumption but not functional correctness.

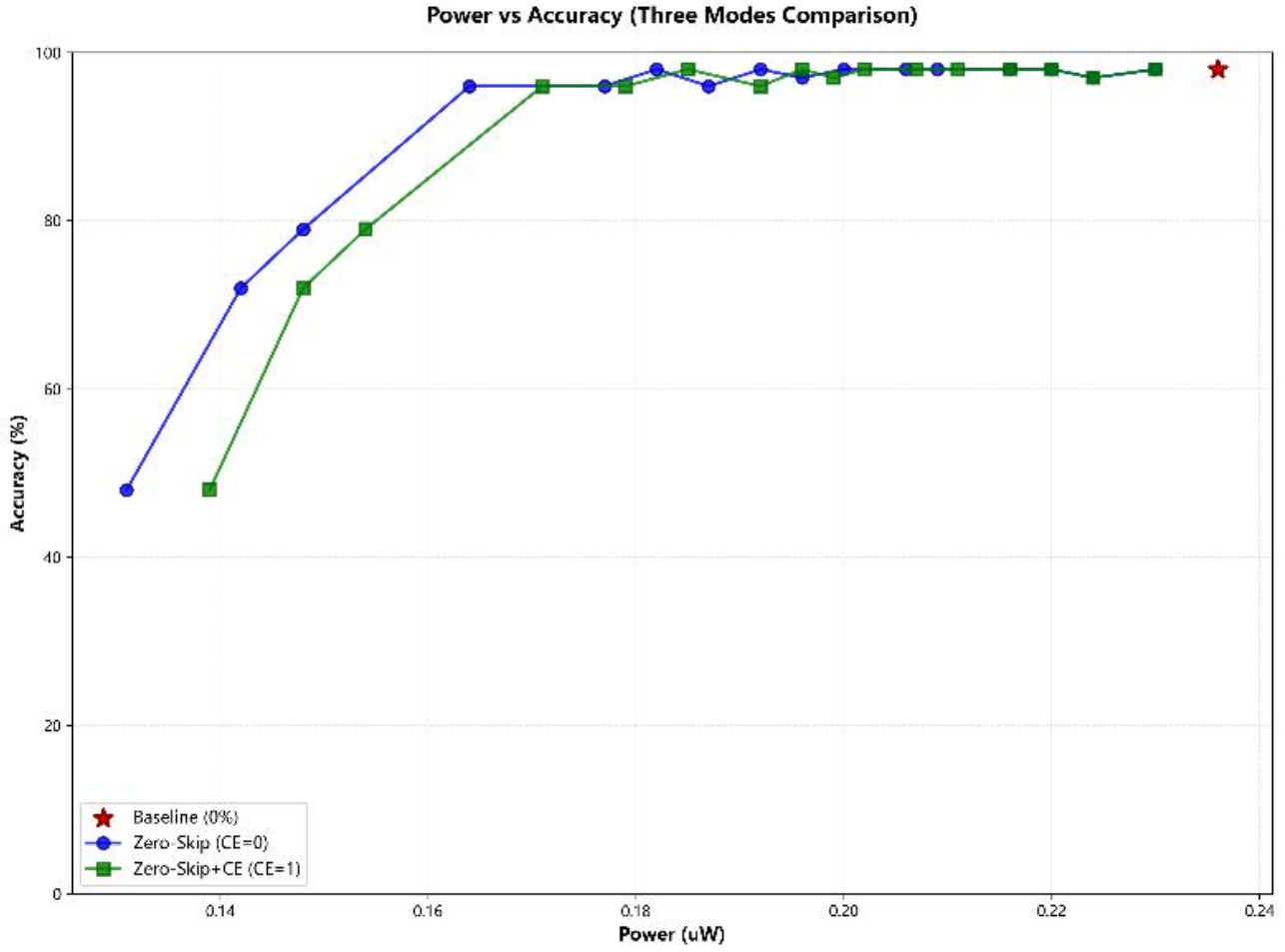

Fig. 2 Relationship between power and accuracy (three modes). The red star denotes the baseline, blue circles denote Zero-Skip, and green squares denote Zero-Skip+CE.

(Picture credit : Original)

4. Discussion

When a small accuracy degradation is acceptable, our results provide practical choices of sparsity. On the evaluation subset formed by the first 500 test images of MNIST, if accuracy is required not to be lower than the baseline 98.0%, a sparsity of 45% can be selected to obtain a 22.9% reduction in Conv1 power without sacrificing accuracy. If at most a 2-percentage-point drop in accuracy relative to the baseline is allowed, a sparsity of 65% can be chosen to achieve a 30.5% reduction in power at the cost of about 2 percentage points of accuracy. These recommendations offer quantitative guidance on the accuracy--power trade-off for inference deployment of neural networks on resource-constrained platforms such as MCUs and FPGAs.

The power-related conclusions in this work are limited to Conv1 rather than the total network power, and thus place more emphasis on data-path and register activities in the early feature-extraction stage. At the same time, CE is defined as data gating rather than integrated clock gating, and the clock network is not turned off. Therefore, most of the potential savings come from suppressing data-path writes and combinational logic toggles, rather than from the clock tree. Under this semantics, we perform gate-level VCD statistics and power conversion with unified parameters and simulation windows, using a commonly adopted approximate power model in digital design to obtain

ISSN 2959-6157

consistent relative comparisons and trend interpretation [10]. It should be emphasized that the dataset and network scale are limited to MNIST and a lightweight CNN, and accuracy is measured on a 500-image evaluation subset of the test set. Hence the conclusions mainly reflect the relative energy-efficiency trends across different sparsity and gating configurations. Extrapolation to more complex tasks, deeper networks, and absolute metrics on the full test set requires further validation and calibration.

Future work will incorporate integrated clock gating (ICG) into synthesis and back-end implementation, quantifying its effect on clock-network toggles and overall dynamic power under timing and clock-tree constraints. On the data-path side, we plan to combine activation zero detection with weight reordering to increase the triggering rate of zero-skipping and write-enable gating, thereby further reducing invalid switching and memory access cost. Based on these extensions, we will broaden the evaluation targets from MNIST and lightweight networks to more complex datasets and deeper models, and conduct board-level power measurements to calibrate the deviation between gate-level simulation and actual power consumption, thus improving the generality and engineering relevance of the conclusions.

5. Conclusion

This paper targets energy-efficiency optimization on resource-constrained platforms and proposes a reproducible experimental flow of "sparsity sweep--VCD counting--power conversion--automatic plotting". Under a unified toolchain and parameter settings, we quantitatively evaluate the accuracy--power trade-offs of zero-skipping at different sparsity levels. The experimental results show that, when Conv1 is used as the evaluation object and the first 500 MNIST test images form the evaluation subset, increasing sparsity from 0% to 65% can reduce power by about 30.5% while maintaining 96% accuracy. If accuracy is constrained not to be lower than the baseline, a sparsity of 45% can be chosen to achieve a 22.9% power reduction while preserving 98.0% accuracy. When a maximum 2-percentage-point drop in accuracy is allowed, 65% sparsity becomes a more aggressive energy-saving option. The comparison of CE shows that, under the implementation where CE acts as data gating and the clock network still toggles, no additional energy savings are obtained in the clock domain. Instead, the switching overhead of the gating-control logic slightly increases power, suggesting that ICG-level clock gating should be introduced and coordinated with zero-skipping in future designs to unlock greater energy-efficiency potential.

Overall, this work provides practical guidelines on sparsity selection and implementation for MCU and FPGA platforms, and establishes a reproducible evaluation baseline for extending the methodology to more complex tasks and deeper network architectures. However, this study is limited by the simplicity of the MNIST dataset, the use of a single network block for power evaluation, and the absence of integrated clock-gating implementations. More comprehensive validations on larger datasets, full-network hardware models, and advanced gating strategies will be explored in future work.

References

[1] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In: Proc. Int. Conf. Learning Representations (ICLR), 2016.

[2] Blalock D, Gonzalez Ortiz J J, Frankle J, Guttag J. What is the state of neural network pruning?. Proc. Machine Learning and Systems (MLSys), 2020, 2, 129-146.

[3] Véstias M P, Duarte R P, de Sousa J T, Neto H C. Fast convolutional neural networks in low density FPGAs using zero-skipping and weight pruning. Electronics, 2019, 8(11), 1321.

[4] Han S, Liu X, Mao H, Pu J, Pedram A, Horowitz M A, Dally W J. EIE: Efficient inference engine on compressed deep neural network. In: Proc. 43rd Int. Symp. Computer Architecture (ISCA), 2016, 243-254.

[5] Parashar A, Rhu M, Mukkara A, Puglielli A, Venkatesan R, Khailany B, Emer J S, Keckler S W, Dally W J. SCNN: An accelerator for compressed-sparse convolutional neural networks. In: Proc. 44th Annu. Int. Symp. Computer Architecture (ISCA), 2017, 27-40.

[6] Aimar A, Mostafa H, Calabrese E, Rios-Navarro A, Tapiador-Morales R, Lungu I A, Milde M B, Corradi F, Linares-Barranco A, Liu S C, Delbruck T. NullHop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. IEEE Trans. Neural Netw. Learn. Syst., 2019, 30(3), 644-656.

[7] Chen Y H, Krishna T, Emer J S, Sze V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In: Proc. 43rd Int. Symp. Computer Architecture (ISCA), 2016, 367-379.

[8] Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, Adam H, Kalenichenko D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, 2704-2713.

[9] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc. IEEE, 1998, 86(11), 2278-2324.

[10] Rabaey J M, Chandrakasan A, Nikolić B. Digital Integrated Circuits: A Design Perspective. 2nd ed. Prentice Hall, 2003.