# A hardware/software co-design vision for deep learning at the edge

**F. Ponzina, S. Machetti, M. Rios, B. W. Denkinger,**
**A. Levisse, G. Ansaloni, M. Peón-Quirós, D. Atienza.**
Embedded Systems Laboratory, EPFL, Switzerland.

***Abstract*—**
**The growing popularity of edgeAI requires novel solutions to support the deployment of compute-intense algorithms in embedded devices. In this article, we advocate for a holistic approach, where application-level transformations are jointly conceived with dedicated hardware platforms. We embody such a stance in a strategy that employs ensemble-based algorithmic transformations to increase robustness and accuracy in Convolutional Neural Networks (CNNs), enabling the aggressive quantization of weights and activations. Opportunities offered by algorithmic optimizations are then harnessed in domain-specific hardware solutions, such as the use of multiple ultra-low-power processing cores, the provision of shared acceleration resources, the presence of independently power-managed memory banks, and voltage scaling to ultra-low levels, greatly reducing (up to 60% in our experiments) energy requirements. Furthermore, we show that aggressive quantization schemes can be leveraged to perform efficient computations directly in memory banks, adopting in-memory computing solutions. We showcase that the combination of parallel in-memory execution and aggressive quantization leads to more than 70% energy and latency gains compared to baseline implementations.**

## INTRODUCTION

The rise and ever-improving accuracy of Artificial Intelligence (AI) is fostering a revolution in a multitude of scenarios, ranging from healthcare to manufacturing. Still, this impressive rise in performance has been fueled by a concurrent increase in complexity [1]. For example, state-of-the-art AI methods for object recognition and automated translation require a workload in the order of GFLOPs[1] for each inference.

Such computational requirements strain the capabilities of digital architectures, especially when considering edge applications where processing is performed entirely or in part at the edge, where devices are typically constrained in terms of computing and memory capabilities. Indeed, a vast number of hardware and software solutions for improving the energy, runtime, and memory efficiency of AI algorithms have been recently proposed [2][3][4]. Nonetheless, hardware and software aspects are often considered in isolation. Instead, we advocate for combining hardware-friendly application optimization strategies and software-friendly architectural solutions to achieve disruptive efficiency gains.

The framework depicted in Figure 1 embodies such a stance. It receives as input a CNN architecture designed (or selected from the state-of-the-art) to achieve the desired classification accuracy

[1]GFLOP: $10^9$ floating-point operations.

 Published by the IEEE Computer Society  

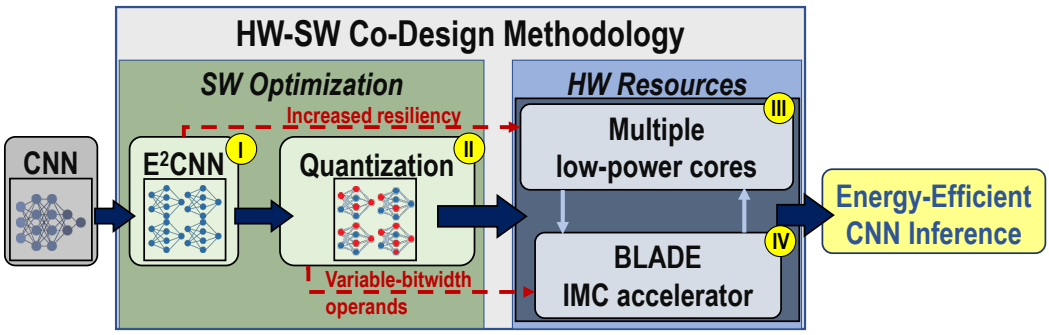


**Figure 1.** Overview of the hardware-software co-design framework. Left: application-level optimizations: ensembling (I) and quantization (II). These are leveraged to drive the design of domain-specific platforms featuring heterogeneous multi-cores (III) and in-memory computing capabilities (IV).

on the target dataset. As for software optimizations (Figure 1.I), we first consider resource-constrained ensembles, which increase accuracy and robustness against sources of internal noise (e.g., memory errors due to sub-nominal operating conditions or approximation due to operands' quantization). Then, this higher resiliency opens the path to aggressive quantization, which reduces memory requirements and improves efficiency (Figure 1.II). Dedicated hardware resources exploit software optimizations. The parallelism exposed by ensembles allows their mapping and execution on platforms featuring multiple ultra-low-power cores (Figure 1.III). Similarly, the presence of multiple, independently power-managed banks opens the opportunity for efficient in-memory computation (Figure 1.IV).

In the rest of the paper, we detail our proposed strategy. We cover software-level optimizations in Section 1. Then, we describe how these can be effectively exploited in the design of domain-specific hardware for edgeAI in Section 2 and Section 3.

## 1 Resource-aware application optimization

Application-level optimization methodologies aim at modifying the structure of CNNs to build models with increased accuracy *and* efficiency.

Toward this goal, the authors of [2] introduced Embedded Ensembles of CNNs ($E^2$CNNs, Figure 1.I). To build $E^2$CNNs, the filters of an untrained CNN architecture are first pruned to obtain a model with lower memory and computing requirements. The obtained structure is

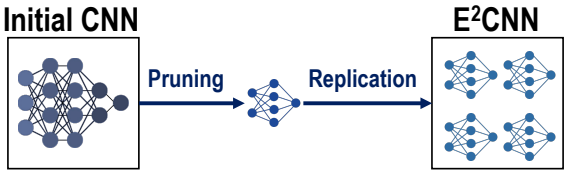


**Figure 2.** In $E^2$CNNs, a CNN model is first pruned. Then, it is replicated several times to build up an ensemble that meets the same memory and computational requirements of the original model.

then replicated, hence deriving models composed of multiple, but lightweight, instances (Figure 2). Afterwards, each instance is independently trained starting from different initial weight values. $E^2$CNNs can also reduce storage requirements when the pruning factor exceeds replication. For example, pruning GoogLeNet by 8x to build an $E^2$CNNs implementation composed of just four instances halves the memory and computational requirements and reduces energy cost by 55%, without any degradation in accuracy when evaluated on the CIFAR100 dataset.

The accuracy and resiliency improvements of $E^2$CNNs support a synergic use of additional optimization approaches. First, the robustness of $E^2$CNNs is exploited by aggressive quantization schemes (Figure 1.II). Indeed, in [3], a strategy is described to aggressively reduces the width of activations and weights in convolutional and fully connected (FC) layers. This approach, summarized in Figure 3, is based on a greedy heuristic that, at each iteration, selects a layer in which the bitwidth should be reduced based on a measure of sensitivity and on its size (since quantizing

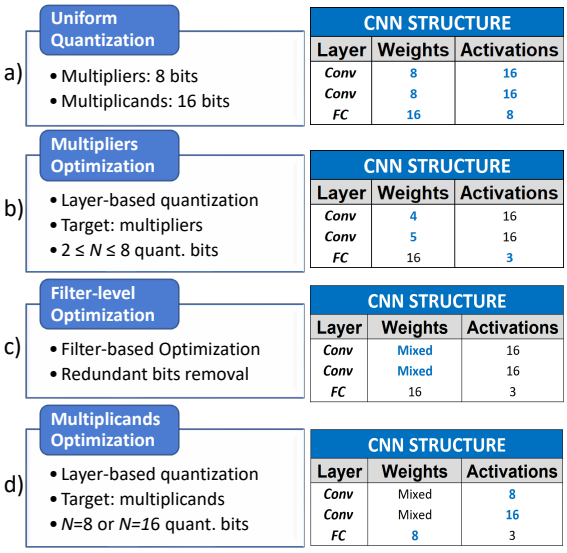


**Figure 3.** Workload-aware quantization and pruning methodology (left). Running example showing how the bitwidth of weights and activations are optimized in different steps (right).

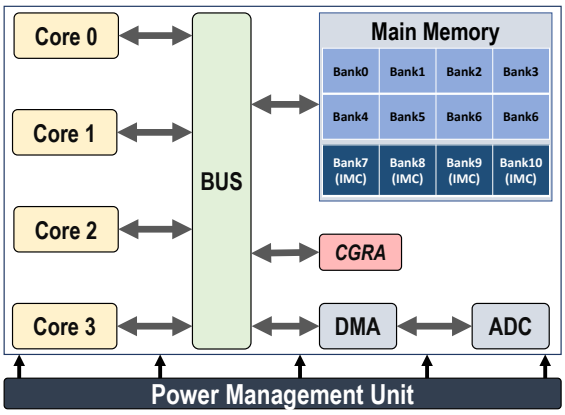


**Figure 4.** Architecture template for EdgeAI platforms, including multiple cores, an independent I/O system, a multi-banked memory supporting in-memory computing (IMC), a reconfigurable CGRA accelerator, and a fine-grained power management unit.

larger layers achieves greater gains). The baseline model (a) is heterogeneously quantized, reducing the bitwidth of weights in convolutional layers and activations in FC layers while meeting a user-defined accuracy level (b). Then, convolutional filters composed of only 0-valued weights are pruned from the model (c), resulting in significant memory and energy savings with no impact on accuracy. Finally, to improve data-level parallelism, the bitwidth of FC weights and convolutional activations is selectively reduced (d). The resulting heterogeneous and fine-grained quantization schemes can be effectively implemented in in-memory computing accelerators, resulting in notable energy gains and very limited accuracy degradations. The energy gains of our approach are discussed in Section 3, where an in-memory computing accelerator supporting the described algorithmic optimization is presented.

Furthermore, ensembles of CNNs exhibit a high degree of robustness towards memory errors, because the instances composing the ensemble exhibit varying weight distributions due to their separate training. Hence, memory errors having a critical impact on the accuracy of one instance may have a significantly lower influence on the others, thus increasing the probability of returning the correct output. The increased resiliency of E$^2$CNNs enables scaling of the supply voltage while tolerating the ensuing error probability when accessing SRAM banks. In [2], experiments on different benchmarks demonstrate that voltage scaling can increase energy efficiency up to 60% without appreciable impact accuracy.

## 2 Domain-specific Hardware

The parallelization of the computing and memory subsystems is key to reducing the energy budget of edgeAI platforms. By using multiple processors, shallower in-order pipelines based on reduced and modular instruction sets (e.g., RISC-V) can be employed in conjunction with dedicated components (e.g., DMAs and accelerators). Such an approach effectively constrains energy without overly sacrificing performance, giving the flexibility to adapt to varying workloads at runtime. For example, when only signal acquisition is performed, solely the ADC components and the DMA transferring data to memory banks are required, while processors and accelerators can be power gated. Moreover, clock gating can be employed to harvest energy-saving opportunities over short time intervals. As an example, cores and accelerators can be clock-gated during synchronization events.

Similarly, dividing the memory into small banks enables energy-saving opportunities. Banks can be individually powered off or put in retention mode when unused, hence increasing effi-

ciency. Moreover, in-memory operations can be supported in multi-banked memories with a high degree of run-time parallelism with limited area overhead, as detailed in Section 3.

A high-level block scheme of an architecture implementing the above-mentioned features is depicted in Figure 4. It features multiple cores to cope with the high workloads of AI applications and several memory banks that can be independently powered off, possibly supporting in-memory computing capabilities. The template architecture also includes flexible coarse-grained reconfigurable arrays (CGRAs), thus enabling the hardware acceleration of computational kernels, as showcased in [12], where energy gains up to 32% are achieved compared to an equivalent single-core system.

Note that hardware-friendly software optimizations presented in Section 1 can efficiently be included in this architecture. CNN instances composing the ensemble can be easily mapped on different cores, which selectively activate memory banks only when needed. The lower workload in each core can then be exploited to reduce the operating frequency (and therefore energy) while abiding to performance constraints, allowing the scaling of the voltage supply.

Although aggressive voltage reduction is possible as digital logic is error-resilient down to the technology voltage threshold, memories (e.g., SRAM cells) usually start failing at higher voltages, hence posing a limit to voltage scaling. The impact of memory errors due to voltage-scaling on CNN accuracy has been studied in [11] and [2], showing that ensembling improves the robustness of CNNs, allowing SRAM memories to operate at sub-nominal voltages while coping with the ensuing errors. These works show energy savings in memories of up to 90% due to voltage scaling while limiting CNN output quality degradation caused by memory errors to just 1%.

The implementation process also plays a role in energy efficiency. Hardware can be optimized at synthesis time by matching the system performance and power consumption to the demands of the target applications using multi-Vt libraries. Such libraries enable low-power and high-performance cells to be instantiated as required to meet timing constraints. Indeed, Figure 5 shows how the normalized energy consumption required to execute a CNN inference varies when different maximum operating frequencies are imposed.

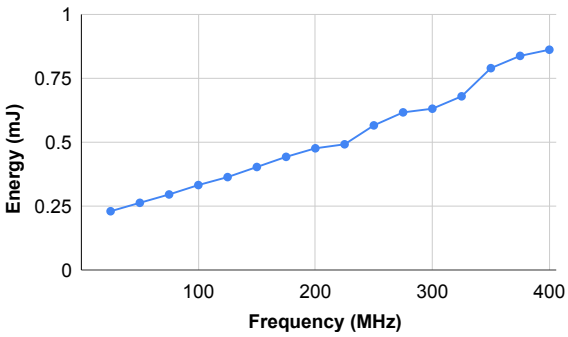


**Figure 5.** Change in the normalized energy consumption of a CNN inference for different frequency constraints imposed during synthesis.

## 3 In-memory computation: BLADE

Enabling computation inside SRAM memory banks is particularly appealing for edgeAI workloads, which are dominated by convolutions or other forms of matrix-matrix and matrix-vector multiplications. The high regularity of these operations in terms of access patterns enables ultra-efficient in-memory computing solutions.

In-memory computing (IMC) architectures can employ technologies ranging from emerging non-volatile memories (eNVM) to traditional CMOS-based memories. IMC based on eNVMs, such as resistive random-access memories (ReRAM), phase change memories (PCM), and magnetic random-access memories (MRAM), can be arranged in cross-points with high integration density. However, these IMC methods rely on non-conventional fabrication processes, complex periphery circuitry including analog-digital converters, and high write currents. On the other hand, IMC using SRAM memories (i) takes advantage of a well-known fabrication process and (ii) can be operated as digital devices, with little additional logic at the periphery of memory cell arrays compared to regular SRAM memories.

Moreover, by relying on SRAMs and due to their very low circuit overhead, SRAM-based IMC architectures can be drop-down replacements for traditional memory banks. Hence, they can leverage the same system-level optimization: they can be power-gated when not used or put in

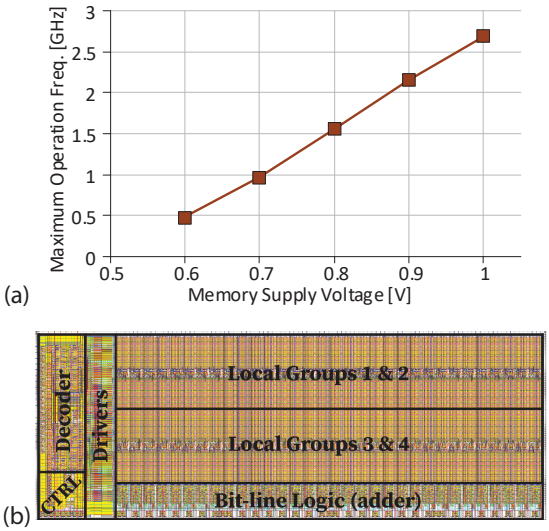


**Figure 6.** (a) Variability-aware performances (GHz) of IMC operations in BLADE simulated in a 28nm CMOS technology. (b) BLADE 2KiB subarray physical layout in 65nm technology as integrated in [5] with the composing blocks highlighted.

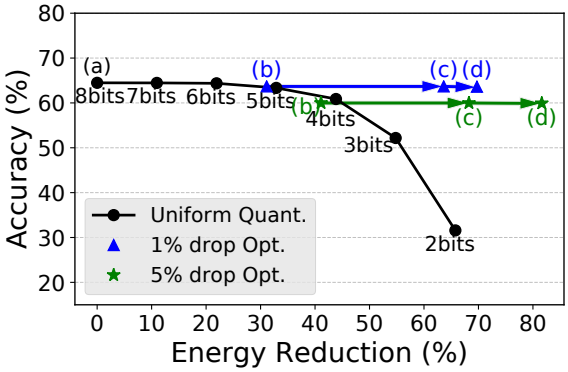


**Figure 7.** Accuracy of MobileNet-v2 [7] on the CIFAR-100 dataset at different energy optimization levels in homogeneously quantized CNNs (black) and our optimized CNNs for a 1% (black) and a 5% (green) user-defined accuracy thresholds. Energy is measured for a BLADE implementation in 28nm CMOS technology. (a)-(d) refer to the optimization steps in Figure 3.

retentive mode when no accesses are performed.

One notable SRAM-based IMC architectural solution is BLADE (Bit-Line Accelerator for Devices on the Edge) [4]. BLADE enables in-situ arithmetic operations and nether rely on analog elements, nor on associated ADCs and DACs. Its circuit-level implementation is compatible with high-density 6T-SRAM bitcells, thanks to an organization of memory cells in Local Groups. Such characteristics make BLADE compatible with a large range of supply voltages and enable an aggressive voltage/frequency scaling, as shown in Figure 6-(a).

In BLADE, operations are performed by simultaneously activating two word lines (WL) of different local groups. IMC operations are performed on the global bit-lines and evaluated by conventional single-ended sense amplifiers. Operations such as additions, subtractions, logic shifts, and bitwise operations can be performed in the memory periphery. By chaining additions and shifts, multiply-and-accumulate (MAC) operations can also be implemented. As convolutional and fully connected layers of CNNs are composed of MAC operations, they can be executed with very high efficiency in a single-instruction multiple-data (SIMD) fashion on the subarrays composing each BLADE bank, as showcased in [3].

BLADE's performance is further increased when low-bitwidth quantization schemes are adopted. Indeed, in SRAM-based IMC architectures, the number of clock cycles required to execute a multiplication is proportional to the bitwidth of the multiplier. Therefore, the application-level strategy described in Section 1 can be effectively harnessed by executing the resulting heterogeneously quantized ensembles in BLADE. Results considering a single-instance implementation are summarized in Figure 7. They show energy (and latency) improvements of 72% with just 1% accuracy degradation compared to a homogeneously 8-bit single-instance CNN.

## 4 Conclusion

In this article, we have discussed the importance of a comprehensive co-design approach for edgeAI, where algorithmic optimizations and hardware architectures are jointly designed. We have shown that very significant energy efficiency gains can be obtained when application-level optimizations are well supported by hardware resources. Embodying this paradigm, we have presented ensembling as a key optimization strategy that improves robustness against aggressive quantization schemes and memory errors. Such characteristics are harnessed by a domain-specific

edgeAI system, which supports parallel execution on multiple ultra-low-power cores, and aggressive voltage-scaling. Additionally, we have shown that heterogeneous quantization CNNs can be effectively leveraged by in-memory computing architectures, and that these can seamlessly integrate into multi-core and multi-banked systems. The presented edgeAI co-design framework achieves up to 60% energy reduction in the memory subsystem thanks to voltage-scaling. Additionally, the in-memory computing accelerator exploits application-level optimizations to improve inference performance and efficiency by 72%, without a significant output quality degradation.

## Acknowledgment

This work has been supported by the ERC Consolidator Grant COMPUSAPIEN (GA No. 725657), the EC H2020 WiPLASH (GA No. 863337), the EC H2020 FVLLMONTI (GA No. 101016776), and the Swiss NSF ML-Edge (GA No. 200020_182009) projects.

## REFERENCES

1. Bianco, Simone, et al. “Benchmark analysis of representative deep neural network architectures.” IEEE Access, 2018.
2. Ponzina, Flavio et al. “E2CNNs: Ensembles of Convolutional Neural Networks to Improve Robustness Against Memory Errors in Edge-Computing Devices.” IEEE Transactions on Computers, 2021.
3. Ponzina, Flavio et al. “A Flexible In-Memory Computing Architecture for Heterogeneously Quantized CNNs.” IEEE Computer Society Annual Symposium on VLSI, 2021.
4. Simon, William Andrew, et al. “BLADE: An in-cache computing architecture for edge devices.” IEEE Transactions on Computers, 2020.
5. http://asic.ethz.ch/2021/Darkside.html, 2021
6. M. Gautschi, et al., “Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices.” in IEEE Transactions on Very Large Scale Integration Systems, 2017.
7. Howard, Andrew G., et al. ”Mobilenets: Efficient convolutional neural networks for mobile vision applications.” arXiv preprint arXiv:1704.04861 (2017).
8. Duch, Loris, et al. “HEAL-WEAR: An Ultra-Low Power Heterogeneous System for Bio-Signal Analysis.” IEEE Transactions on Circuits and Systems I, 2017.
9. P. Davide Schiavone, et al., “Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications.” 27th International Symposium on Power and Timing Modeling, Optimization and Simulation, 2017.
10. Pullini, Antonio, et al., “Mr.Wolf: An Energy-Precision Scalable Parallel Ultra Low Power SoC for IoT Edge Processing.” in IEEE Journal of Solid-State Circuits, 2019.
11. Denkinger, Benoît W. et al., “Impact of Memory Voltage Scaling on Accuracy and Resilience of Deep Learning Based Edge Devices.” in IEEE Design & Test, 2020.
12. De Giovanni, Elisabetta et al., “Modular Design and Optimization of Biomedical Applications for Ultra-Low Power Heterogeneous Platforms.” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020.

**Flavio Ponzina** received the M.Sc. degree in Computer Engineering from Politecnico di Torino, Italy, in 2018. He is currently a PhD student at the Embedded Systems Laboratory (ESL), EPFL. His main research interests include low power architectures and AI-based systems optimization.

**Simone Machetti** received the M.Sc. degree in computer engineering from the Politecnico di Torino, Turin, Italy. He worked as a firmware engineer at SPEA, Volpiano, Italy, a world-leading company in the design and manufacture of Automatic Test Equipment. He is currently pursuing a Ph.D. degree in electrical engineering at the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. His research interests include hardware and software co-design for ultra-low-power embedded devices and artificial intelligence algorithms for the Internet of Things.

**Marco Rios** received the M.Sc. degree in Computer Science and Electronics For Embedded Systems from Université Grenoble Alpes, Grenoble, France, in 2018. currently a PhD student at the Embedded Systems Laboratory of EPFL, Switzerland. His research interests include design of integrated systems and circuits, in-SRAM computing and the system impact of emerging memories.

**Benoît Walter Denkinger** received his M.Sc. in robotics and autonomous systems from the Institute of Electrical and Micro Engineering, EPFL. He is currently pursuing a Ph.D. degree with the Embedded Systems Laboratory (ESL), EPFL, Switzerland. His research interests include low-power architectures for biomedical applications and artificial intelligence (AI)-

enabled Internet-of-Things (IoT) devices.

**Alexandre Levisse** received his Ph.D. degree in Electrical Engineering from CEA-LETI, France, and from Aix-Marseille University, France, in 2017. From 2018 to 2021, he was a post-doctoral researcher in the Embedded Systems Laboratory at the Swiss Federal Institute of Technology Lausanne (EPFL). From 2021, he works as a scientist in EPFL. His research interests include circuits and architectures for emerging memory and transistor technologies as well as in-memory computing and accelerators.

**Giovanni Ansaloni** is a researcher at the Embedded Systems Laboratory of EPFL (Lausanne, Switzerland). He previously worked as a Post-Doc at the University of Lugano (USI, Switzerland) between 2015 and 2020, and at EPFL between 2011 and 2015. He received a PhD degree in Informatics from USI in 2011. His research efforts focus on domain-specific and ultra-low-power architectures and algorithms for edge computing systems, including hardware and software optimization techniques.

**Miguel Peón-Quirós** received a Ph.D. on Computer Architecture from UCM, Spain, in 2015. He collaborated as a Marie Curie scholar with IMEC (Leuven, Belgium) and as postdoctoral researcher with IMDEA Networks (Madrid, Spain). He has participated in several H2020 and industrial projects and is currently a postdoctoral researcher at EPFL. His research focuses on optimizations for embedded devices.

**David Atienza** is Professor of EE, and heads the Embedded Systems Laboratory (ESL) at EPFL, Switzerland. His research focuses on design methodologies for edge AI in the context of Internet of Things (IoT) and thermal- and energy-aware design for server architectures and datacenters. He has published more than 350 publications on these topics, and is an IEEE Fellow and an ACM Distinguished Member.