Impact of memory voltage scaling on accuracy and resilience of deep learning based edge devices

Impact of Memory Voltage Scaling on Accuracy and Resilience of Deep Learning Based Edge Devices

Benoît W. Denkinger, Flavio Ponzina, Soumya S. Basu, Andrea Bonetti, Szabolcs Balási, Martino Ruggiero, Miguel Peón-Quirós, Davide Rossi, Andreas Burg, David Atienza

***Abstract*—Energy consumption is a significant obstacle to integrate deep learning into edge devices. Two common techniques to curve it are quantization, which reduces the size of the memories (static energy) and the number of accesses (dynamic energy), and voltage scaling. However, static random access memories (SRAMs) are prone to failures when operating at sub-nominal voltages, hence potentially introducing errors in computations. In this paper we first analyze the resilience of artificial intelligence (AI) based methods for edge devices—in particular convolutional neural networks (CNNs)—to SRAM errors when operating at reduced voltages. Then, we compare the relative energy savings introduced by quantization and voltage scaling, both separately and together. Our experiments with an industrial use case confirm that CNNs are quite resilient to bit errors in the model, particularly for fixed-point implementations ($5.7\,\%$ accuracy loss with an error rate of 0.0065 errors per bit). Quantization alone can lead to savings of up to $61.3\,\%$ in the dynamic energy consumption of the memory subsystem, with an additional reduction of up to $11.0\,\%$ introduced by voltage scaling; all at the price of a $13.6\,\%$ loss in accuracy.**

***Index Terms*—Fault-tolerance, neural nets, energy-aware, yield analysis.**

I. Introduction

Deploying artificial intelligence (AI) capabilities on edge devices is important to increase their autonomy, reduce latency and solve privacy issues. However, devices at the edge face important constraints in terms of performance and energy. AI tasks place high demands both on performance and energy consumption, particularly in the case of deep learning algorithms such as convolutional neural networks (CNNs) for object identification and classification. Therefore, finding new ways of reducing energy consumption during inference is vital for the successful deployment of AI on edge devices.

CNNs have shown high resilience to imprecision [1]. In particular, fixed-point quantization is an optimization technique that exploits this resilience to reduce the size of the operands and of the arithmetic operations from a classic 32-bit floating point format to a more compact fixed-point representation. Previous works have shown that value widths of 16, 8, 4 or even less bits retain sufficient accuracy for most practical cases. The advantage of quantization in the context of this work is that it reduces both the memory footprint of the networks, which reduces the static energy consumption linked to leakage current, and the number of memory accesses, which in turn reduces the dynamic energy consumption. For example, an 8-bit fixed-point quantization reduces the memory footprint of a CNN by a factor of 4 compared to the 32-bit floating point version, while also reducing the number of memory accesses by a similar factor—because CNNs process values mostly consecutively and the processor can access four consecutive values with a single 32-bit memory read or write.

Voltage scaling is another method to reduce energy consumption, at the expense of reduced frequency. However, although logic elements tolerate well reductions in operating voltage, static random access memories (SRAMs) are more sensitive and start experiencing errors sooner. The more voltage scaling is applied, the more errors appear in SRAMs.

Process variations across SRAM bitcells and chips make each individual bitcell vulnerable at a different voltage level. For a given size and voltage, all the chips have (statistically) the same number of errors, but their concrete distribution along memory cells is random. Therefore, their effect at the application level is both potentially relevant and unpredictable, as some parts of the generally error-resilient CNN models (and of the intermediate results) may be more susceptible to errors than others. To explore the interaction of chip variability and CNN resilience, we perform simulations over a complete set of populations consisting of hundreds of chips. Then, instead of analyzing the resulting average accuracy, we perform a detailed yield analysis that uncovers the trade-offs between energy consumption, desired accuracy and chip yield [2]. As a result, the main contributions of this paper are the following:

- An analysis of the resiliency of CNNs to SRAM errors produced when working at sub-nominal voltages.
- An assessment of the dynamic energy savings in the memory subsystem of an edge device produced by quantization and voltage scaling.
- A method to determine the yield of a set of manufactured devices given a target energy budget and CNN accuracy.

The rest of this paper is organized as follows. First,

This work has been partially supported by the ERC Consolidator Grant COMPUSAPIEN (GA No. 725657), the ML-Edge RTD project supported by the Swiss NSF under Grant 200020_182009, and the EC H2020 RECIPE project (GA No. 801137).

B. Denkinger, F. Ponzina, S. Basu, M. Peón-Quirós and D. Atienza are with the Embedded Systems Lab (ESL), EPFL, Lausanne, Switzerland.

A. Bonetti and A. Burg are with the Telecommunication Circuits Laboratory (TCL), EPFL, Lausanne, Switzerland.

S. Balási and M. Ruggiero are with Nespresso, Romont, Switzerland.

D. Rossi is with the Integrated Systems Laboratory, ETH Zürich, Switzerland, and also with the Department of Electrical, Electronic, and Information Engineering, University of Bologna, Italy.

in Section II we review the existing literature on related resilience and accuracy exploration techniques. Then, in Section III we explain our methodology to evaluate the resilience of CNNs against SRAM errors. Next, in Section IV we present our case study and analyze the results obtained in Section V. Finally, Section VI summarizes our conclusions.

II. Background

A. Convolutional neural networks

In this work we consider CNNs, a specialized type of deep neural network (DNN) that expects data with specific temporal or spatial structure as input, such as time-series or images. These constraints allow the encoding of certain specific properties into the network architecture, making CNNs particularly suitable for computer vision tasks [3].

However, the exceptional performance of CNNs comes at the cost of high computational complexity and memory requirements, which makes their deployment and real-time inference on embedded edge devices a challenging task. CNNs, and DNNs in general, perform millions to billions of Multiply-Accumulate (MAC) operations and memory accesses. However, as a memory access costs more energy than a MAC operation, reducing the amount of energy per access is an effective way of reducing the overall energy consumption. For example, in the PULPissimo system, a RISC-V microcontroller for ultra-low power IoT applications, SRAMs can consume up to $71\,\%$ of the total energy when running a binarized neural network [4]. Therefore, memory subsystem optimizations are key to achieve energy-efficient deep learning on embedded devices.

B. CNN resilience

CNNs are known for their robustness against noise. On the one hand, weight quantization [5], which can be seen as a type of noise, has only a small impact on accuracy while reducing both the number of memory accesses and the memory size. On the other hand, aggressive voltage scaling, which can lead to additional energy savings [6], introduces errors in the normal operation of SRAMs. Thus, in this work we explore as well the resilience of CNNs against this new type of errors.

The topic of CNN resilience against memory errors is currently attracting attention in the field of edge devices system design. For example, preliminary results on the impact of relaxed programming conditions with resistive memories on the accuracy of binarized neural networks (BNNs) are reported in [7]. The authors of [8] have proposed a framework to test the resilience of different DNN architectures with quantization under a certain bit error rate. In comparison with these works, here we explore the resilience of quantized non-binary CNNs—as our test case requires a higher accuracy than possible with BNNs—against memory errors introduced by voltage scaling and quantization with the goal of improving energy efficiency in a concrete industrial use case. Additionally, we conduct a yield analysis to model the behavior of a population of chips with errors in concrete memory cells.

TABLE I
Bit-error probability for a commercial 6T-SRAM manufactured on a $40\,\mathrm{nm}$ CMOS process and operated at decreasing (sub-nominal) voltages [10, Table 3].

Voltage (V)	P(bit error)	Voltage (V)	P(bit error)
0.85 (nominal)	0	0.65	0.0007
0.75	1.3×10^{-5}	0.60	0.0022
0.70	0.0001	0.55	0.0065

III. Methodology

A. Yield analysis considering error-resilience

The bit-flip errors of an SRAM operated below the critical supply voltage are stochastically distributed within the bitcell array because of their large dependence on cell-to-cell process variations [9]. Thus, even if the probability of a bit-flip error per bitcell is defined for a voltage point, the positions of bit-flip errors as well as their impact on the application are different for each fabricated die. In fact, the potential degradation on the application does not only depend on the number of bit-flip errors in an SRAM, but also on their position within the the bitcell array. For example, errors on the most-significant bits (MSBs) are more likely to impact the quality of service (QoS) compared to errors appearing on the least-significant bits (LSBs). Given the different bit-flip error pattern in a population of fabricated memories, it is crucial to evaluate the error resilience of the considered benchmarks for each of the produced dies [2]. Moreover, it is mandatory to assess the measured benchmark quality for a population of dies with a *yield analysis* [2], as other methods based on either averaging or on the worst measured quality do not capture the characteristics of a population of dies or are too pessimistic, respectively.

The yield analysis on the error resilience of applications with faulty memories is enabled by an evaluation process divided in three stages, as shown in Fig. 1. First, the **Model preparation** phase generates the information about bit-flip errors for a population of dies (i.e., number and position of each bit-flip error in the die) from a fault model that uses bit-flip error statistics from silicon measurements as input. In this work, we refer to the information on the bit-flip errors for each fabricated memory as *error mask*. Second, during the **Full system emulation** phase, two nested loops are used to evaluate the error resilience of a benchmark CNN across different dies where the outer loop iterates through the error masks (one for each die), while the inner loop iterates over different input data for the CNN and evaluates the output quality of each case. Third, the **Analysis** phase is the assessment on the error resilience of the CNN. It is conducted on the cumulative distribution function (CDF) of the measured quality for yield analysis.

B. Simulation with SRAM errors injection

The increasing requirements for low-power operation in modern applications have driven the necessity for modelling memories working at low voltage supplies. In that regard,

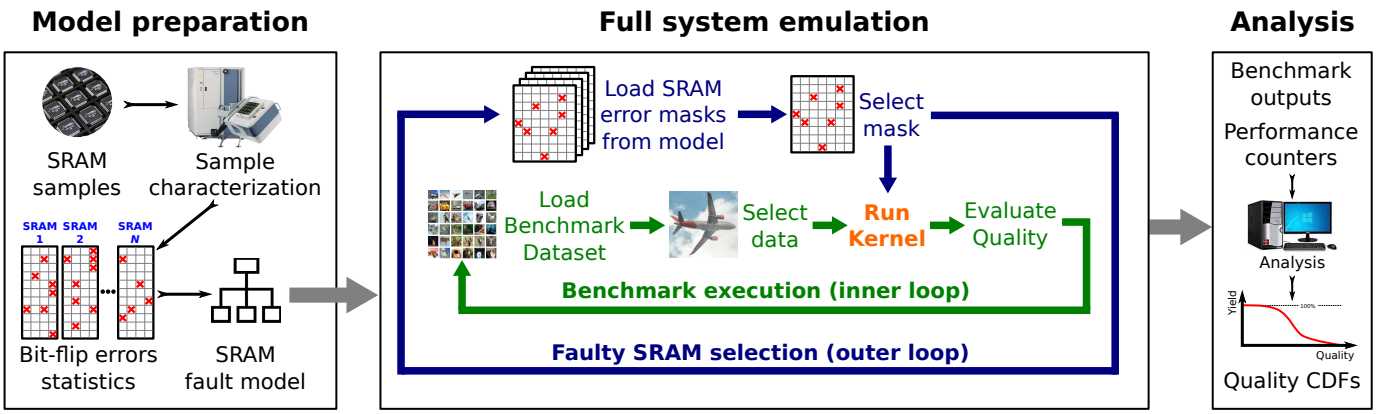


Fig. 1. Emulation platform and flow for quality of service (QoS) and performance assessment [2].

[10] presents a characterization study on the bit error rate for a 6T SRAM manufactured on a 40 nm CMOS process across different (sub-nominal) voltage levels. We base our experiments on their measurements, as reproduced in Table I.

Our error injection simulation process uses an instrumentation mechanism based on C++ templates. This allows to easily simulate the memory subsystem, injecting errors in specific bit cells and counting the number of read/write accesses to get at the end an estimation of the energy requirements. In particular, we introduce a memory class that forces "stuck-at-0" and "stuck-at-1" bit masks in its words as needed. Then, we use a set of wrapper classes for variables of different sizes, corresponding to the data widths used in the different CNN implementations. These wrapper classes provide overloaded assignment, copy and cast operators. In this way, the same code basis of the final system can be used with minor changes for experimentation on a server farm to simulate thousands of chips working at different voltage levels. Despite the instrumentation and the different architectures involved, the number of memory accesses measured during simulation corresponds within a close margin to the number of memory accesses measured in the final platform using the standard *perf* tool.

Given a memory size and voltage level, the number of errors present in each mask is computed by multiplying the number of bits in the SRAM by the error probability given in Table I. The exact positions of the errors in the mask are randomly calculated using a uniform distribution.

To assess the accuracy of the different CNN implementations, we first compute the accuracy of the CNN on our dataset without error injection. This situation corresponds to running the application at nominal voltage. Then, we run the CNN at a given sub-nominal voltage level and measure the resulting accuracy. Subsequently, for each voltage level, we repeat this step for 1000 different error masks, thus simulating a whole population of manufactured chips at different voltage levels. The impact of different error masks—even with the same number of errors—on accuracy depends on the specific error locations, as shown in Section V.

C. Model and data placement

To exhaustively study the resilience of CNNs, we initially considered three different scenarios: 1) All the application data are loaded in the voltage-scaled SRAM. 2) Only the model parameters (weights, biases) are loaded in the voltage-scaled SRAM, whereas the buffers used by the internal layers are stored in a safe SRAM (working at nominal voltage). 3) The model is stored in the safe SRAM, while the buffers are stored in the voltage-scaled SRAM.

Our initial experiments have shown that CNNs accuracy drops dramatically even for relatively small numbers of errors in the buffers. In contrast, CNNs are significantly more resilient to errors in the model itself. However, these findings are in contrast with the results shown in [8], where the activation buffers are more resilient to error than the model. The explanation is that the studied industrial implementation (cf. Section IV) stores values in the activation buffers using an integer and a decimal part, without minimizing the number of bits devoted to the representation of the integers. In particular, half of the representation is used to represent the decimal part and the other half is used to represent the sign bit and the integer part. In [8], on the other hand, the integer part is reduced to the minimum number of bits required to run the inference. As a consequence, larger errors might arise in our experiments because errors in the MSBs of the integer part have a strong influence, particularly for activations with small absolute values. In contrast, the model weights lack an explicit integer part while stored in memory. Therefore, in the rest of this work we consider only the second scenario.

IV. Experimental setup for the target industrial case study

A. Description of the CNN architecture

In this work, we consider an industrial application for capsule recognition in an AI-enabled coffee machine. The goal is to perform real-time image classification with a target recognition time lower than 100 ms. The desired accuracy is 95 %, with a minimum acceptable accuracy of 90 %. The machine recognizes the capsule when the user inserts it and suggests different options based on the capsule type. The used CNN (Fig. 2) is similar to the AlexNet network, namely, each

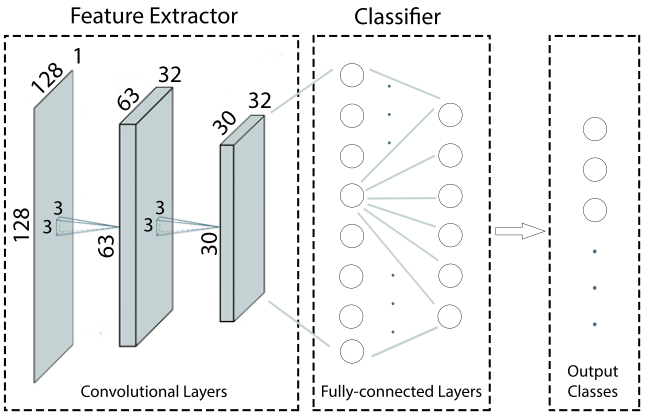


Fig. 2. Architecture of the CNN used in our industrial case study.

TABLE II
NUMERIC REPRESENTATIONS AND THEIR CORRESPONDING MEMORY CONSUMPTION AND ACCESSES FOR THE COMPLETE CNN.

	Number of bits			Memory Size[1]	read/write
	Sign	Int.	Dec.	(MiB)	accesses
FP_32	–	*N/A*	–	4.0	1.06×10^8
FXP_8_16	1	8	7	1.6	5.54×10^7
FXP_4_32	1	28	3	2.6	6.95×10^7
FXP_4_8	1	4	3	0.8	4.07×10^7

convolutional layer is followed by a Rectified Linear Unit (ReLU) and a max pooling layer. However, the implementation is optimized specifically for our application. Hence, the last fully-connected layers classify the images into 15 classes.

B. Description of the classifier

To achieve real-time classification, we developed an optimized C/C++ inference engine. We implemented two versions: a 32-bit floating point version (FP_32) and a customizable fixed-point version (FXP_i_j). We derived several fixed-point implementations with varying numbers of bits to store the model parameters (i) and the intermediate buffers or computations (j). For example, in FXP_8_16 the model weights are stored on 8 bits and the computations are done (and stored) on 16 bits. All the fixed-point versions use a 2's complement representation. The model parameters are fully decimal, that is, $weight_i \in (-1, +1)$. We tried to developed a special version of the FXP_8_16 to increase the resilience of the CNN where the two upper bits of the integer part in the buffers are masked and replaced by the value of the sign bit. The underlying observation is that the intermediate results in our case study take values in the range $(-64, +64)$. The additional two MSBs are not needed and, when subject to bit flips at reduced voltages, can cause large changes in the absolute value of the data. This masking process is feasible with simple bit operations. However, initial experiments have shown that it behaves similarly to the original FXP_8_16 version. The

[1]Total memory size of the model and the buffers.

TABLE III
ENERGY CONSUMPTION PER ACCESS FOR A 16 KiB SRAM BUILT ON A 40 nm CMOS PROCESS AT DIFFERENT VOLTAGE LEVELS (pJ/access).

	850 mV	**750 mV**	**700 mV**	**650 mV**	**600 mV**
Read	9.447	7.572	6.766	6.047	5.416
Write	5.868	4.703	4.202	3.756	3.364

explanation is that even if activations may theoretically assume values in the range $(-64, +64)$, in our industrial use case, most of them are very close to zero: hence, to fully protect the integer part from large errors, more than two MSBs should be masked to effectively increase the resilience of the model. To verify this hypothesis, we conducted additional experiments in which we protected all the integer bits from possible errors. At the lowest voltage (i.e., 600 mV), the accuracy drop was lower than 6 %, while the unprotected version was affected by more than a 23 %. As protecting the complete integer part would require more advanced techniques, a better solution may be to consider non-symmetric fixed-point representations with fewer bits in the integer part, as suggested in [8].

The memory footprint of the different implementations and their respective number of memory accesses are reported in Table II. Quantization has an immediate effect on the CNN footprint, which translates into reductions of both the static and dynamic energy consumption of the system. For example, FP_32 requires 1.6 MiB to store the model (weights and biases) and 2.4 MiB for the buffers, which represents a total of 4.0 MiB. In comparison, FP_4_8 requires only 0.2 MiB for the model and 0.6 MiB for the buffers, hence reducing memory footprint by a factor of 5. Moreover, it performs 2.6 × less memory accesses than the FP_32 version, thus reducing considerably the dynamic energy. Consequently, quantization is normally advisable for edge devices.

To compensate the accuracy degradation caused by quantization (i.e., fixed-point representation), we introduce an additional training step that maintains two sets of parameters during training: a quantized version and a full precision one, as proposed in [11].

V. EXPERIMENTAL RESULTS

We conduct our analysis considering that the model is stored in an unreliable (voltage-scaled) SRAM whereas the buffers are stored in a reliable memory. As mentioned before, our preliminary analyses showed that, if the buffers are also stored in the voltage-scaled memory, the accuracy of the CNN degrades very quickly as the number of errors increases. Therefore, we carefully evaluate the impact of this growing number of errors in the final application output quality.

The first result of our analysis is that the floating point representation is very sensible to errors. In addition to changes in the magnitude of the fractional and exponent parts, as well as on the bit sign, the IEEE-754 representation imposes specific rules on the format of valid numbers. Therefore, certain bit flips may alter the representation in such a way that it is not valid anymore, being interpreted instead as a Not-a-Number (NaN) value. NaNs do not only affect the current

TABLE IV
CNN ACCURACY AND MEMORY SUBSYSTEM DYNAMIC ENERGY CONSUMPTION AT DIFFERENT VOLTAGE LEVELS. $850\,\mathrm{mV}$ REPRESENTS THE MAXIMUM ACCURACY AT NOMINAL VOLTAGE.

	850 mV		**750 mV**		**700 mV**		**650 mV**		**600 mV**	
	Accuracy (%)	**Energy (µJ)**	**Accuracy (%)**	**Energy (µJ)**	**Accuracy (%)**	**Energy (µJ)**	**Accuracy (%)**	**Energy (µJ)**	**Accuracy (%)**	**Energy (µJ)**
FP_32	99.8	976.4	33.8	889.2	–	851.6	–	818.4	–	789.0
FXP_8_16	99.8	511.9	99.8	482.9	99.4	470.4	95.3	459.3	76.6	449.6
FXP_4_32	95.0	634.3	95.0	615.0	94.9	606.7	93.7	599.3	89.6	592.9
FXP_4_8	92.7	378.3	92.7	359.0	92.7	350.8	90.9	343.4	86.2	336.9

computation, but they also propagate to the following ones. In order to avoid this situation, we evaluated the detection of NaNs and substituted them with a known value, specifically 0, to limit their impact, but without success.

A. Analysis of accuracy and energy consumption

In order to determine the dynamic energy consumption of the SRAM with the different numeric representations and working at varying voltage levels, we use the values presented in Table III, which were measured for a $16\,\mathrm{KiB}$ SRAM; we use multiple banks to build bigger memories as needed. Table IV shows the average accuracy and dynamic energy consumption of the SRAM for the different numeric representations (quantizations) and voltage levels considering 1000-chip populations. In this work, we omit the static energy consumption produced by leakage current. The reason is that the voltage levels that we consider are enough to produce significant energy savings in the SRAMs, but not to make leakage the dominant factor in energy consumption. A preliminary estimation places the contribution of leakage in the range of $2\,\%$ to $5\,\%$ for most cases.

Clearly, quantization has the biggest impact on energy consumption, mainly because it reduces the number of memory accesses proportionally to the reduction in memory footprint (see Table II). In particular, using FXP_8_16 instead of FP_32 reduces the dynamic energy consumption by $47.6\,\%$ at nominal voltage level with virtually the same accuracy. For a slight accuracy loss of $7.1\,\%$, FXP_4_8 can achieve an energy reduction of $61.3\,\%$.

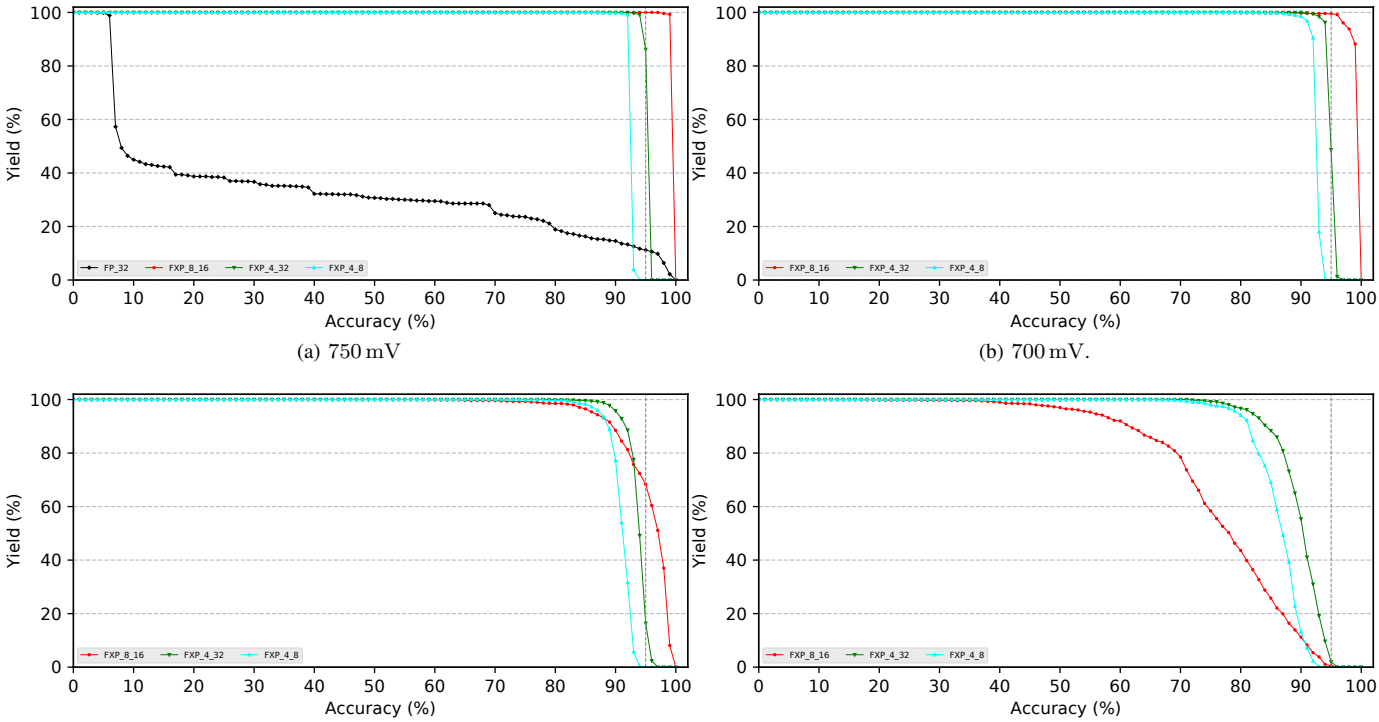
Voltage scaling provides additional reductions in energy consumption at the expense of additional accuracy loses, which may be moderate in some cases. For example, applying voltage scaling after quantization saves an additional $8.1\,\%$ of dynamic energy at $700\,\mathrm{mV}$ for FXP_8_16 and $7.3\,\%$ for FXP_4_8, with no significant accuracy loss. In addition, if energy consumption is optimized further, then large accuracy losses are observed. In this context, the FXP_4_8 version with memory operating at $600\,\mathrm{mV}$ for the model is the best case, with a $61.3\,\%$ reduction coming from the quantization and another $10.9\,\%$ reduction thanks to voltage scaling. However, these savings come with an impact on accuracy of $13.6\,\%$. The aforementioned savings are additional to the energy saved by the system processors operating themselves at lower voltage points, a factor that is not taken into account in our numbers.

B. Yield analysis

As explained in Section III, the accuracy numbers reported in Table IV, which are an average for all the chips in the studied populations, do not reflect the real-world objectives of system design and manufacturing with a high yield of chips that achieve a guaranteed minimum quality. Inside a population, some of the chips will have very poor quality performance, but most devices may be usable for a given quality criterion. Yield is strongly related to the cost per chip as a higher yield leads to lower costs per chip and vice-versa. Fig. 3 shows the yield at different voltages for the different implementations, that is, the percentage of chips that would meet a specific QoS at each voltage level. In this case, the yield analysis confirms the poor resilience of the floating point (FP_32) version: Fig. 3a shows that only $\approx 14\,\%$ of the chips would meet a QoS of $90\,\%$ at this voltage. Therefore, voltage scaling is not an option if a floating point representation is needed. With respect to the fixed point representations, while the FXP_8_16 versions are the best for the two highest voltages compared to the FXP_4_8/32 versions, the inverse holds for the lowest voltages. The reason is that, in general, smaller model representations with just the required numbers of bits for the integer part lead to better resilience in presence of voltage scaling.

Figure 3 shows also that using only the average accuracy to evaluate a population of chips is not enough. For example, FXP_4_8 has an average accuracy of $86.2\,\%$ when working at $600\,\mathrm{mV}$. However, a complete yield analysis unveils that less than $12\,\%$ of the chips can achieve an accuracy higher than $90.0\,\%$ at that voltage level. Also, Table IV shows that FXP_8_16 working at $650\,\mathrm{mV}$ achieves similar average accuracy than FXP_4_32. However, Fig. 3c shows quite different yields. The yield analysis allows the designer to determine the number of chips that will meet a certain level of accuracy with a maximum energy consumption. Conversely, it enables classifying the chips according to different energy efficiency (minimum voltage levels) for a desired minimum accuracy.

Using the yield information, Table V allows us to observe that $100\,\%$ of the chips will be able to meet the desired accuracy of $95\,\%$ for this industrial case working at nominal voltage in all numeric representations except for FXP_4_8; indeed, that specific configuration cannot meet the requirements in any case. Then, operating at $750\,\mathrm{mV}$, $100\,\%$ of the chips will still be able to meet that requirement using FXP_8_16, whereas only $86\,\%$ will meet it using FXP_4_32 (only $11\,\%$ of the chips will be able to reach $95\,\%$ accuracy at this voltage level



(a) 750 mV (b) 700 mV.

(c) 650 mV. (d) 600 mV.

Fig. 3. Yield-accuracy trade-off for a thousand chips at varying voltages. Activations in "safe" SRAM; CNN model in voltage-scaled SRAM.

TABLE V
PERCENTAGE OF CHIPS (YIELD) THAT CAN MEET THE DESIRED ACCURACY OF 95 % FOR THIS INDUSTRIAL CASE.

	850 mV	**750 mV**	**700 mV**	**650 mV**	**600 mV**
FP_32	100.0	11.2	0.0	0.0	0.0
FXP_8_16	100.0	100.0	99.5	66.3	0.2
FXP_4_32	100.0	86.2	48.7	16.4	2.0
FXP_4_8	0.0	0.0	0.0	0.0	0.0

using floating point). At 700 mV, more than 99 % of the chips will still meet the requirement using FXP_8_16, but only 48 % will meet it using FXP_4_32. Finally, at 650 mV, only 68 % of the chips will achieve 95 % accuracy with FXP_8_16 and 16 % using FXP_4_32. Less than 2 % of the chips will be able to meet 95 % accuracy at 600 mV, with any version.

VI. CONCLUSIONS

In this paper, we have explored the impact of on-chip memory (i.e., SRAM) voltage scaling on the precision and resilience of edge devices relying on CNNs. Our experiments show that the CNN used in our industrial case is quite resilient against errors on the model, while the intermediate buffers, which hold the activations, are more critical. Although our experiments were limited to a specific application, we believe that our results can be generalized to similar CNNs based on the AlexNet architecture.

We have exploited this information to explore, with an accurate SRAM error-injection framework, the accuracy loss that the CNN experiences after the introduction of quantization and SRAM voltage scaling (only for the model) in the context of an industrial application with the goal of reducing energy consumption in the memory subsystem. Our results show that quantization is the most efficient method, with energy savings up to 61.3 % for accuracy losses as small as 7.1 %. Voltage scaling can be used to achieve further reductions on energy consumption, albeit with a higher impact on accuracy. Additionally, our experiments have shown that floating-point representations should be avoided in sub-nominal voltage contexts. Moreover, our experiments agree with previous work, showing that minimizing the number of bits used by the integer part should minimize the magnitude of noise introduced in the model, therefore increasing its robustness. In that line, an interesting experiment is exploring how error injection affects the CNN accuracy if it is taken into account during the training step of the model.

Finally, we have shown how to perform yield analysis to evaluate the impact of voltage scaling at system-level on a real industrial application. Our results indicate that it is crucial to characterize precisely the information about the percentage of chips that can achieve acceptable accuracy when subject to SRAM errors. This yield analysis enables a realistic trade-off between yield and QoS. In our experiments, we have shown how to perform a cross energy-accuracy-yield analysis for commercial CNN-based edge devices.

REFERENCES

[1] I. Goodfellow *et al.*, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[2] A. Widmer *et al.*, "FPGA-Based Emulation of Embedded DRAMs for Statistical Error Resilience Evaluation of Approximate Computing Systems," in *DAC*, 2019.

[3] A. Krizhevsky *et al.*, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.
[4] F. Conti *et al.*, "XNOR Neural Engine: A Hardware Accelerator IP for 21.6-fJ/op Binary Neural Network Inference," *IEEE TCAD*, vol. 37, 2018.
[5] B. Moons *et al.*, "Minimum energy quantized neural networks," *ACSSC*, 2017.
[6] B. Reagen *et al.*, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," *ISCA*, 2016.
[7] T. Hirtzlin *et al.*, "Outstanding bit error tolerance of resistive RAM-based binarized neural networks," *CoRR*, 2019. [Online]. Available: https://arxiv.org/pdf/1904.03652.pdf
[8] B. Reagen *et al.*, "Ares: a framework for quantifying the resilience of deep neural networks," in *DAC*, 2018.
[9] T. Gemmeke *et al.*, "Resolving the memory bottleneck for single supply near-threshold computing," in *DATE*, 2014, pp. 1–6.
[10] D. Bortolotti *et al.*, "Approximate compressed sensing: Ultra-low power biosignal processing via aggressive voltage scaling on a hybrid memory multi-core processor," in *ISLPED*, 2014, pp. 45–50.
[11] M. Courbariaux *et al.*, "BinaryConnect: Training Deep Neural Networks with binary weights during propagations," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.00363

Benoît W. Denkinger received the M.Sc. degree in Robotics and Autonomous Systems from the Institute of Electrical Engineering, EPFL, Lausanne, Switzerland, in 2017. He is currently a PhD student at the Embedded Systems Laboratory (ESL), EPFL. His main research interests include low power architectures for bio-medical applications and AI-enabled IoT devices.

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.

Soumya S. Basu received the Ph.D. in Electrical Engineering from the Embedded Systems Laboratory at EPFL, Lausanne, Switzerland. He is currently working as a hardware engineer at Thales Suisse SA, Zurich. His research interests are on low-power reconfigurable architectures and IoT devices.

Andrea Bonetti obtained the Ph.D. degree in 2019 at the Telecommunications Circuits Laboratory (TCL), EPFL, Lausanne, Switzerland. From 2012 to 2014, he worked as Design Engineer at AMS AG, Rapperswil, Switzerland for the development of interfaces in MEMS sensors. In 2016, he visited the Circuits Research Laboratory at Intel Labs, Hillsboro (OR), USA for the development of low-power circuits. His research activity is focused on energy-quality scaling techniques and memory design for low-power computing.

Szabolcs Balási received the M.Sc. degree in Electrical Engineering from the Institute of Electrical Engineering, EPFL, Lausanne, Switzerland. He is currently Electronic Engineering Manager at Nestlé Nespresso SA. His interests include HW & SW co-design for embedded systems, machine learning and edge computing in IoT devices.

Martino Ruggiero obtained the Ph.D. degree from the University of Bologna, Italy, in 2009. He is currently Head of Systems Architecture and Special Projects at Philip Morris International in Lausanne, Switzerland. His research interests include low-power architectures for embedded and AI-enabled IoT devices.

Miguel Peón-Quirós received the Ph.D. degree on Computer Architecture from the Complutense University of Madrid, Spain, in 2015. He is currently a postdoctoral researcher at EPFL. His research interests include energy and memory optimizations for embedded systems, and AI-enabled IoT devices.

Davide Rossi received the Ph.D. degree in Electronics Engineering from the University of Bologna, Italy, in 2012. He is an Assistant Professor at the Energy Efficient Embedded Systems Laboratory at the University of Bologna. His current research interests include ultra-low power multicore SoC design and applications.

Andreas Burg Andreas Burg (S'97-M'05) received the Dr. sc. techn. degree from the Integrated Systems Laboratory of ETH Zurich, in 2006. In 2007 he co-founded Celestrius, an ETH-spinoff in the field of MIMO wireless communication. In January 2009, he joined ETH Zurich as SNF Assistant Professor. In January 2011, he joined EPFL, where he is leading the Telecommunications Circuits Laboratory. He was propoted to Associate Professor with Tenure in June 2018. He is currently an editor of the Springer Journal on Signal Processing Systems, the MDPI Journal on Low Power Electronics and its Applications, and of the IEEE Transactions on VLSI. He is also a member of the EURASIP SAT SPCN and of the IEEE TC-DISPS and the CAS-VSATC.

David Atienza is associate professor of electrical and computer engineering, and the director of the Embedded Systems Laboratory (ESL) of EPFL, Switzerland. He received an ERC Consolidator Grant in 2016, and was the recipient of the IEEE CEDA Early Career Award in 2013, and the ACM SIGDA Outstanding New Faculty Award (ONFA) in 2012. He is an IEEE Fellow and a Senior Member of ACM. His research interests include system-level design and thermal-aware optimization methodologies and ultra-low power system architectures for wireless body sensor nodes.