Energy-Aware Adaptive Multi-Exit Neural Network Inference Implementation for a Millimeter-Scale Sensing System

Yuyang Li, Yawen Wu, Xincheng Zhang, Jingtong Hu, *Senior Member, IEEE*, and Inhee Lee, *Senior Member, IEEE*

***Abstract*— Implementing a neural network (NN) inference in a millimeter-scale system is challenging due to limited energy and storage size. This article proposes an energy-aware adaptive NN inference implementation that utilizes one of two exits with different accuracies and computation options. The early-exit path provides a shorter processing time but less accuracy than the main-exit path. To compensate for the reduced accuracy, it additionally applies the main-exit path if the entropy of the early-exit inference is higher than a predetermined value. The NN is implemented with a custom low-power 180-nm CMOS processor chip and a 90-nm embedded flash memory chip and tested by the CIFAR-10 dataset. The measurement results show that the implemented convolutional NN (CNN) reduces processing time and thus energy consumption by 43.9% compared with a main-exit-only method while sacrificing its accuracy from 69.9% to 66.2%. Also, we explore the required minimum battery capacity at each optimal configuration for accuracy and/or energy consumption to achieve energy-autonomous operation under measured exemplary light profiles. It requires a minimum battery capacity of 855 mJ, acceptable for the target miniature system with two millimeter-scale batteries (684 mJ each). Compared with the state-of-the-art CNN technique (BranchyNet) allowing early stopping, the proposed design improves the accuracy by 0.7% and 3.3% to maintain energy-autonomous operation with two and one millimeter-scale batteries, respectively. Compared with the state-of-the-art lightweight CNN technique (MobileNet), this work provides flexibility with a tradeoff between accuracy and processing time for different application requirements.**

***Index Terms*— Battery capacity, energy harvesting, miniature system, multiexit, neural network (NN).**

I. Introduction

RAPID evolution in computing systems has dramatically reduced their form factor over 70 years from a room-size mainframe to a miniature Internet-of-Things (IoT) device with centimeter-/millimeter-scale size [1]. The miniature systems become attractive monitoring tools in a variety of applications, including ecological, biomedical, security, and infrastructure [2]–[5]. However, there are still substantial challenges in miniature system design, including limited memory size and battery capacity. For applications where raw data size is significant, it is not efficient to store the original data in small memory in the miniature system. For instance, a small image with 32 × 32 pixels, RGB color channel, and 8-bit color depth takes 3 kB (e.g., CIFAR-10 dataset [6]). A 128-kB flash memory [7], designed for a miniature system, stores only 42 images. To store key information from a greater number of images, it is more efficient to analyze the raw images and save only the necessary information in the memory. For example, the convolutional neural network (CNN), one of the recently highlighted machine learning (ML) techniques, categorizes the CIFAR-10 dataset into ten classes so that the final output can be stored in 4 bits [8]. A CNN with 70-kB coefficients enables the system to store both the coefficients and 116 000 inference results in the 128-kB memory.

The ML technique performs intensive computation and consumes considerable energy. Thus, implementing it in a miniature system encounters an issue with limited energy resources since the system size constraint restricts physical battery size and thus battery capacity. For example, a thin-film battery with 5.7 mm × 6.1 mm stores an only electrical charge of 50 μAh or energy of 684 mJ [9]. A low-power TI MSP430 processor can discharge the battery in 4.2 days just by its standby power of 0.5 μA [10]. Moreover, it is very difficult to replace a battery once the millimeter-scale system is fully encapsulated for physical protection [11]. Hence, a miniature system usually employs an energy harvesting technique to recharge the battery using environmental energy and extends the system lifespan [11]–[13]. However, the small battery capacity limits usable energy consumption before harvesting typically weak or intermittent environmental energy and charging the battery again. Here, energy consumption should be minimized not to fully discharge the battery since the continuous operation is desirable in typical applications.

On-device neural network (NN) processing has been actively researched for devices with limited resources in energy and memory [14]–[17]. One promising energy reduction technique, which can be efficiently applied to a millimeter-scale system, is a multiexit NN. Wu *et al.* [18], Gobieski *et al.* [19], Teerapittayanon *et al.* [20], and

Manuscript received May 27, 2021; revised October 25, 2021, January 20, 2022, and March 13, 2022; accepted April 22, 2022. Date of publication May 11, 2022; date of current version June 29, 2022. This work was supported in part by the University of Pittsburgh Hewlett International Grants Program, NSF CNS-2007274, and the University of Pittsburgh Center for Research Computing through the resources provided. *(Corresponding author: Inhee Lee.)*

The authors are with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: inhee.lee@pitt.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2022.3171308.

Digital Object Identifier 10.1109/TVLSI.2022.3171308

Huang *et al.* [21] proposed a multiexit approach by adding branches with additional exits. It saves processing time and energy consumption by sacrificing accuracy to an acceptable level from a shallower path. However, the multiexit network cannot be applied to our target millimeter-scale system as it is since their networks do not simply fit to significantly limited resources (e.g., 50 μAh battery and 128 kB memory) and usable energy highly depends on environmental energy.

This work explores a way to implement a multiexit CNN modified from [18] in a miniature platform and demonstrates the feasibility of perpetual operation under a measured natural light profile, as an extended work of [22]. It employs two exit options such as the main and early exits and chooses one according to available energy in a battery. The early-exit path provides a shorter processing time but less accuracy than the main-exit path. To compensate for the reduced accuracy, it additionally applies the main-exit path if the entropy of the early-exit inference is higher than a predetermined value. Using the entire test set of the CIFAR-10 dataset, the measurement result shows that the implemented CNN reduces processing time and thus energy consumption by 43.9% compared with the main-exit-only method while decreasing the accuracy from 69.9% to 66.2% (only 3.7% reduction). The simulation using actual measured light intensity data demonstrates a long-term operation situation without battery replacement. It shows an energy-autonomous operation for 1.5 months, covering the worst case energy in two miniature thin-film batteries. This work also describes how to optimize the battery energy and entropy thresholds for the implemented network for different use scenarios such as the minimum required accuracy, the minimum required average processing time, and the figure of merit (FoM) combining accuracy and required battery capacity together. Compared with the state-of-the-art CNN technique (BranchyNet) [20] allowing early stopping, the proposed design improves the accuracy by 0.7% and 3.3% to maintain energy-autonomous operation with two and one millimeter-scale batteries, respectively. Compared with the state-of-the-art lightweight CNN technique (MobileNet) [17], this work provides flexibility with a tradeoff between accuracy and processing time for different application requirements.

An important difference in implementing and processing an NN of the millimeter-scale system in this work, from other IoT or edge-level systems associated with the tight energy budgets, is to take battery capacity and thus its size into account. This work finds the best configurations for a given goal (e.g., accuracy and/or energy consumption) and computes the minimum battery capacity required to be energy autonomous from a given environmental light profile. The approach enables us to find the optimal configurations by including acceptable battery capacity or the maximum battery size as a factor, which is critical for the total system form factor. Typical IoT or edge-level systems are not limited by battery size or their batteries are easy to be replaced after a reasonable lifetime. Hence, it is not critical for them to include the battery capacity and size in implementing and processing an NN.

A miniaturized system can be designed without energy storage by directly powering its circuits using an energy harvester or wireless power transfer [23]–[24]. The pure energy harvesting system performs a given task only when a harvesting unit provides enough power to load circuits. The system does not save excessive energy even if harvested power is larger than the power consumption of the load circuit. On the other hand, a millimeter-scale system, envisioned in this work, stores extra harvested energy in a miniaturized battery, and it powers the system using the stored energy when harvested power cannot support the load circuits due to weak available energy in the environment. Thus, energy consumption must be optimized, instead of power consumption, to achieve perpetual operation.

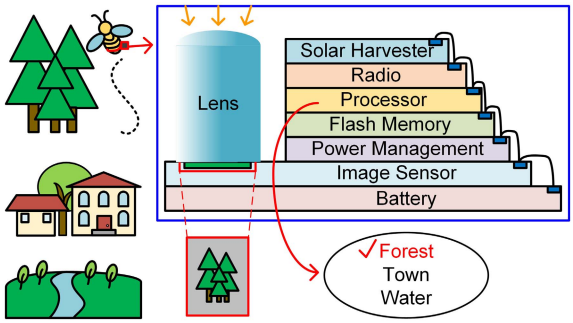

Fig. 1. Proposed system and an example application.

This article is organized as follows. Section II introduces the target millimeter-scale system. Section III explains how to implement a multiexit CNN in the system, and Section IV discusses its scalability. Section V discusses the experiment results, and finally, Section VI concludes the work.

II. Target System

Fig. 1 shows a target system for an example application where a miniature smart imager is attached to a flying insect [25], [26] to study its living environment. The system takes its surrounding pictures, extracts their key information, and saves the inference results (e.g., forest, water, and town) in memory. Once it comes near a gateway, the collected data are wirelessly retrieved. The miniature imager mainly consists of three functional blocks for image sensing, wireless communication, and data processing and storage. In developing such a system, CNN implementation is one of the major challenges, including how to attach the system to a flying insect, harvest light energy, and communicate wirelessly. In this article, we focus on the CNN implementation with limited resources for data processing and storage functions.

A die-stacking structure [1] has been used to develop a variety of millimeter-scale systems with different sensing modalities [4], [27]–[33], including imagers. The platform consists of multiple thinned die stacked vertically, as shown in Fig. 1. The structure provides the maximum functionality (or silicon area for integrated circuits) per unit volume. The modular platform enables us to optimally miniaturize systems for different applications. The layers are connected by bonding wires that form a low-power data bus [34] and deliver regulated supply voltages. The target system includes chips, such as a light energy harvester [11], a processor [1], a flash memory [7], a radio transceiver [35], a power management unit (PMU) [36], and a battery [9].

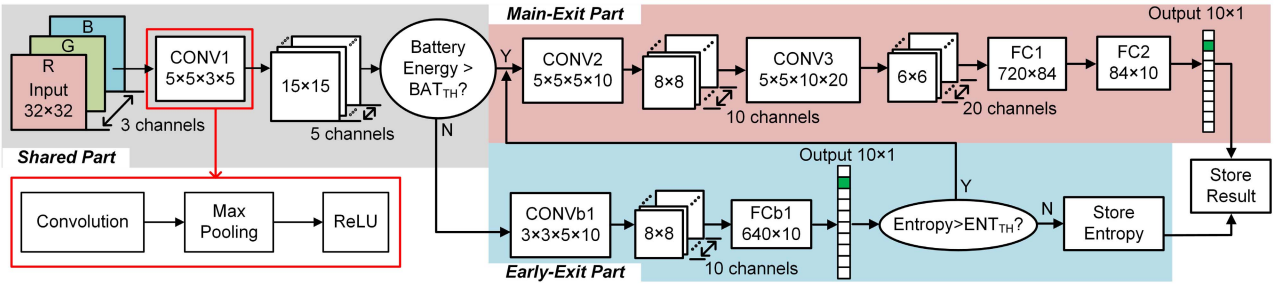

Fig. 2. Proposed CNN structure.

To extend the system's operational lifetime, a light energy harvester is included in the miniature system. Photovoltaic (PV) cells convert light energy to electrical charge in a small dimension [37]. Different energy harvesting solutions can also be considered for the system. For example, a piezoelectric transducer can convert mechanical vibration into electrical energy, but its typical size exceeds an acceptable millimeter scale [38]. An antenna can convert RF energy into electrical energy [39], but it is not always available in the target field application.

The processor includes a commercial ARM Cortex-M0 processor and a custom low-power 16-kB SRAM, fabricated in a 180-nm CMOS process [1]. NN processors (neural processing units (NPUs), NN cores, and so on), providing acceleration for neural processing, are well studied in recent years and have been integrated into processors [40]. Due to the size limitation of the target system, the accelerator can be implemented only with a more advanced process, but the leakage current of SRAM should be maintained at a similar level so as not to increase system power consumption in sleep mode. The 128-kB embedded flash memory is designed with custom low-power peripheral circuits and fabricated in a 90-nm NOR flash process [7]. The nonvolatile memory enables the system to keep the stored inference results and neural network coefficients even when the system loses power from extremely low-energy conditions.

III. Proposed CNN Inference Implementation

Similar to a typical NN, the multiexit CNN requires memory space to store the coefficients. The CNN needs to be compact to reduce the required memory space in the target system. Fig. 2 shows the proposed CNN designed for the target system based on [18], which processes images compatible with the CIFAR-10 dataset. The depth of the NN and the number of channels are determined by considering limited memory and battery capacity to process the images with few data movements. It mainly consists of three blocks such as the shared, early-exit, and main-exit parts. The input is an image with 32 × 32 pixels, RGB channels, and 8 bits. Each convolutional block includes a convolutional layer with a filter size of 5 × 5 or 3 × 3, a max-pooling layer, and a rectified linear unit (ReLU). Starting from CONV1, the system processes each layer in order. After CONV1, PMU checks available energy. When available energy is lower than a threshold ($\mathrm{BAT_{TH}}$), it performs the shallower network using the early-exit path.

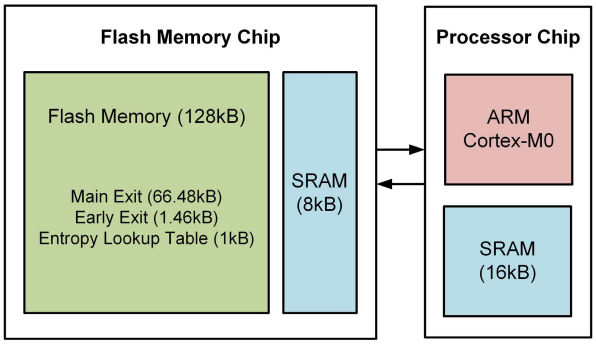

Fig. 3. Implemented CNN with processor and flash memory chips.

The two exits have the same importance, so their average loss is minimized during training. The coefficients of the model occupy 73.2 kB of storage, which is 57% of a 128-kB memory. To compensate for its reduced accuracy from the shallower network, it computes entropy as a confidence level. If the entropy level is lower than a threshold ($\mathrm{ENT_{TH}}$), the inference result is stored in the memory; otherwise, it aborts the result and executes the main-exit part instead. Here, the entropy is expressed as [18]

$$p_i = e^{\mathrm{output}_i} / \sum e^{\mathrm{output}_i} \tag{1}$$

$$\mathrm{Entropy} = -1 \cdot \sum p_i \cdot \log_2 p_i. \tag{2}$$

The proposed NN is implemented with the custom processor and flash memory chips, as shown in Fig. 3. They are designed for low-power miniature systems and fabricated in 180-nm CMOS and 90-nm embedded flash processes, respectively [1], [7]. The two chips communicate through a low-power bus [34]. The processor chip receives image data and runs the NN by loading coefficients for each NN layer from the flash memory chip. The coefficients include the weights and bias for the convolution and fully connected layers and the lookup table (LUT) for entropy calculation, taking 68 kB in the flash memory. The weights and bias are trained through the Pytorch framework and then quantized to 8-bit fixed-point values. Each layer outputs 16-bit fixed-point results.

Fig. 4(a) shows the measured time taken across layers for the main- and early-exit paths. At the processor clock frequency of 3.2 MHz, processing an image takes 4.07 min for the main-exit path but only 1.27 min for the early-exit path, reducing the processing time and thus energy consumption

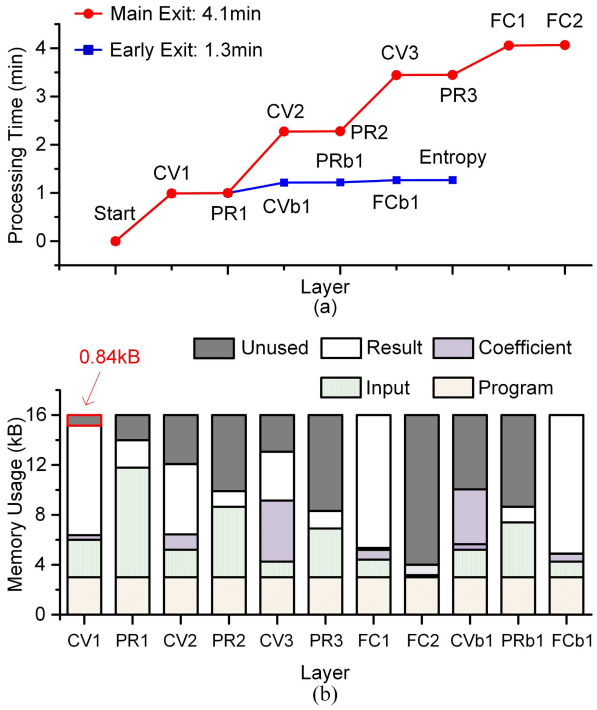

Fig. 4. CNN layer. (a) Measured processing time. (b) Simulated memory usage.

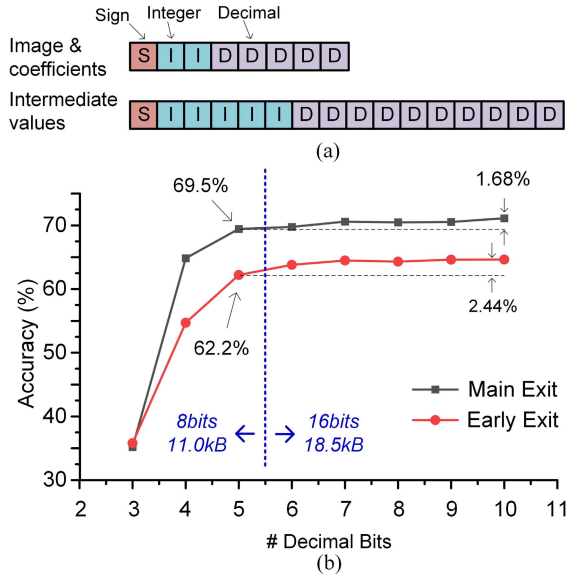

Fig. 5. Simulated CNN for 10 000 images with different bit allocations. (a) Bit field of the input and intermediate data. (b) Accuracy and memory space required.

by 68.8%. Computing the convolutional layer dominates the total processing time due to multiples of matrix multiplication in series. The first convolutional layer (CV1) costs a longer time than the other following layers since the input size becomes smaller at later layers due to the pooling operation. Fig. 4(b) shows simulated allocated memory usage for different layer computations. The minimum remaining SRAM memory of 0.84 kB in the processor chip can store 1720 4-bit inference results, and additional results can be stored in the flash memory.

Wu *et al.* [18] used more advanced compression algorithms to the multiexit CNN for higher accuracy, including reinforced-learning-assisted, nonuniform pruning, and quantization. However, the work has been done targeting a commercial microprocessor for a centimeter-scale edge device, so the same technique cannot be applied. Thus, this work indispensably keeps the code simple and the data unified, considering the limited memory size and the hardware only supporting 8-, 16-, and 32-bit fixed-point computation. The weights and bias values are uniformly quantized to the same format as well as the input image.

Fig. 5(a) presents the bit assignment for the input and intermediate data, which is designed in 8 and 16 bits, respectively. To efficiently utilize the 32-bit SRAM, the intermediate value, multiplication of two 8-bit values, is not truncated. Keeping the 16-bit result increases the accuracy by 0.6% and takes more memory space from 11 to 18.5 kB. Fig. 5(b) shows how the inference accuracy changes according to the number of decimal bits of the input image and layer coefficients. The decimal part is set to 5 bits to reduce the data from 16 to 8 bits and hold the data of one layer fully in the SRAM while

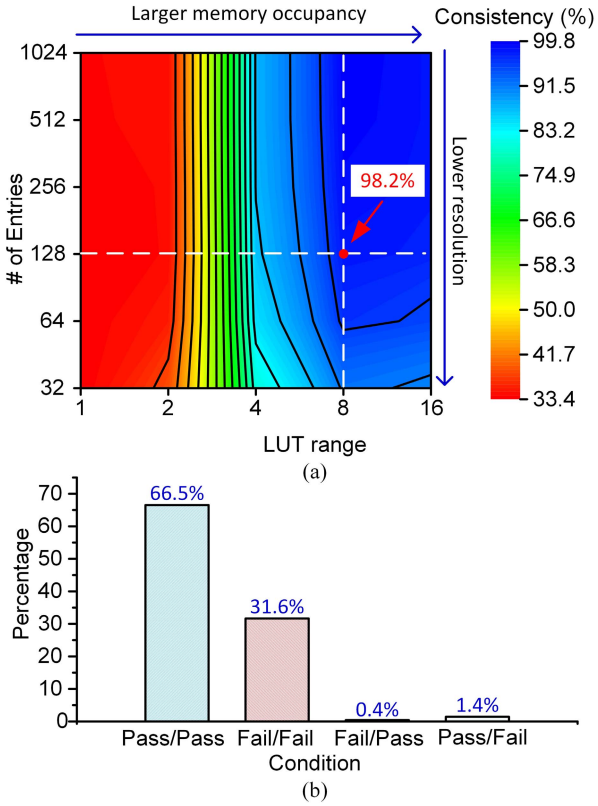

Fig. 6. Simulated LUT with ENT_{TH} of 1.9. (a) Consistency between LUT-based and actual entropy across the LUT range and the number of entries. (b) Histogram of the consistency at the chosen LUT design parameters (LUT-based/actual entropy).

sacrificing the accuracy by only 1.69% and 2.44% for the main- and early-exit paths, respectively.

After processing the early-exit path, the design calculates the entropy of the output to judge its credibility.

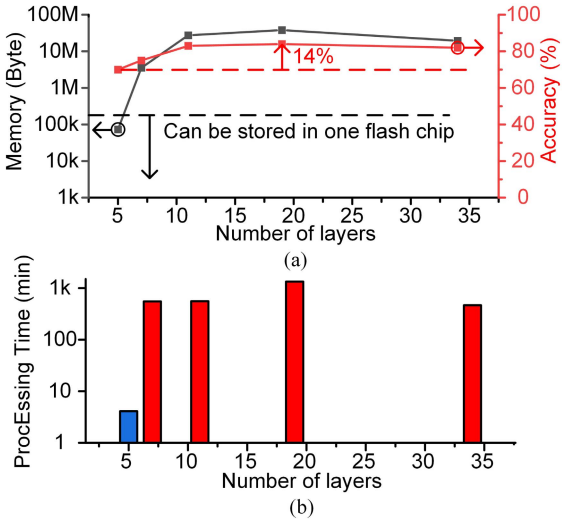

Fig. 7. Simulated resources for CNNs across the number of layers. (a) Estimated accuracy and memory size. (b) Estimated processing time.

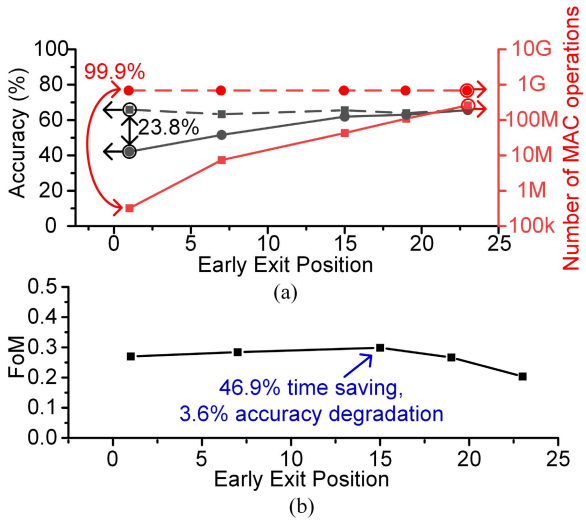

Fig. 8. Simulated resources for 34-layer CNN with different early exits. (a) Estimated accuracy and number of MAC operations. (b) FoM.

The exponential and logarithmic functions are implemented by LUTs. The logarithmic function uses a 128-entry LUT stored in the flash and the input range from 0.0071825 (1/128) to 1. The boundary values are used if the input is out of the range. Fig. 6(a) shows the consistency between the LUT-based and actual entropy values, compared with ENT_{TH} of 1.9, as the input range and number of entries change. Increasing the input range gives more accurate lookup numbers for extremely large or small values while reducing the resolution. Increasing the number of entries improves the resolution issue while leading to larger memory occupancy. For inconsistency, less than 2%, the 128 entries and input range from -8 to 8 are chosen. Fig. 6(b) presents the simulated consistency with the chosen design parameters; 0.6% of the failed cases (higher than ENT_{TH}) with the actual entropy values pass (lower than ENT_{TH}) with the LUT-based method. On the other hand, 4.3% of the passed cases with the actual entropy values fail with the LUT-based method.

IV. Scalability of CNN Implementation

In addition to the designed five-layer CNN, we investigate whether a larger CNN can be implemented in the system, using Python and floating-point computation. Fig. 7(a) shows the estimated accuracy and required memory size and Fig. 7(b) shows the estimated processing time for the CFAR-10 dataset across different numbers of layers for a CNN from 5 to 34. Although the accuracy becomes higher with more numbers of layers, the required memory size increases relatively faster. Even the seven-layer CNN requires a 3.5-MB flash memory chip in the target system and takes 9.2 h to complete the processing, which cannot be acceptable in a typical application. Moreover, the 34-layer CNN needs a 19-MB flash memory chip and takes 7.8 h.

Also, we investigate how the multiexit benefits for the 34-layer CNN as an example of a large NN by assuming that a target system has more hardware resources (e.g., memory size and available energy) than our target. Here, we use a more complex dataset, CFAR-100, to distinguish the effect of different exit points better. Fig. 8(a) shows the simulated accuracy and the calculated number of multiply–accumulation (MAC) operations for an early-exit path branched at different positions of the original 34-layer path. Fig. 8(b) shows an FoM to visualize the optimal point that maximizes the multiplication of the estimated relative processing time saving and the average accuracy as an example. The relative processing time is calculated by assuming that a system uses the early-exit path for half of the inputs due to lack of available energy and the main path for the other half, and then, it is normalized to the processing time of the main-exit path only method. From this study case, the CNN with an early exit branched from the middle (15th layer) of the main exit shows the best FoM. Please note that the processing time is proportional to energy consumption in the condition that instantaneous power is relatively constant.

In future research, a systematic way can be explored to find an optimal exit from an NN and constraints. For a given NN with a large number of layers, we create a new exit with information such as a branching point and additional layers. From this exit, we find accuracy, processing time (or energy consumption), and required memory size and evaluate them as a target FoM considering their different significance. Then, finding branching points and additional layers to achieve the best FoM becomes an optimization problem. There are two potential issues to process this optimization problem. First, there are a significant number of candidates for the additional layers after a branching point for a new exit. It is not only a different number of additional layers but also the complexity of each layer. A smaller set of candidates needs to be chosen in a certain way to simplify the optimization problem. Second, finding accuracy for all the candidates can take significant effort. We can reduce the computation of accuracy of the candidates by saving data from each potential branching point when processing the original main-exit path and applying it to the newly created additional layers. Computation effort for this

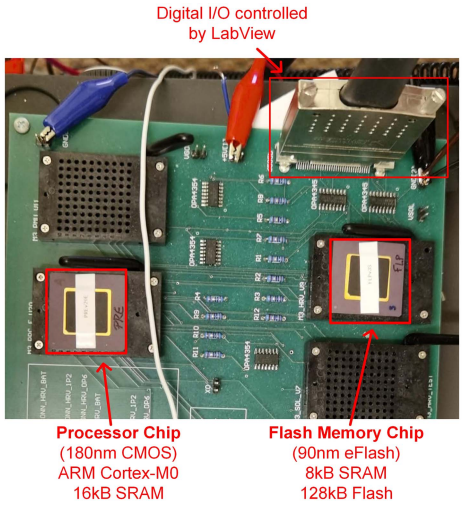

Fig. 9. Photograph of the chips and testing setup.

work can be relieved by reducing the number of candidates (as suggested in the first issue) or finding simpler equations that predict accuracy.

V. Experiment Results

Fig. 9 shows the photograph for the testing setup. The program code for the NN operation and its coefficients are written to the SRAM of the processor chip or the flash memory by the National Instruments PCIe-6535b digital I/O device. Inference and entropy results are recorded by monitoring the bus communication using the same device. The power consumption is measured by Keithley 2401/2450 source meters.

Fig. 10(a) presents the measured entropy distribution of the early-exit path using 10 000 images from the test set of the CIFAR-10 dataset. The average entropy of correct inferences is 1.27, while that of incorrect ones is 1.89. Fig. 8(b) shows the inference accuracy when the results are accepted if entropy is lower than a threshold (ENT_{TH}). At extremely low ENT_{TH}, there are rarely incorrect results, leading to high accuracy. However, only 2% of the results are kept as the final output. At ENT_{TH} of 1.9, 65.0% of the result is accepted without running the main-exit path additionally, among which 73.4% are correct. We can improve the accuracy by running the main-exit path when the entropy is higher than ENT_{TH}, with increasing the total energy consumption. Fig. 10(c) and (d) shows the accuracy and number of correct inference results, respectively. After computing the main-exit path additionally, the accuracy at ENT_{TH} of 1.9 is 66.2%. Since all the results are accepted, the number of correct inferences increases from 4786 to 6620. Fig. 8(e) shows the average processing time across ENT_{TH} in this approach. Note that the total time for inference, when entropy > ENT_{TH}, is the processing time of "early-exit path + main-exit path – shared part." At ENT_{TH} of 1.9, it reduces processing time and thus energy consumption by 46.2% (from 67.5 to 36.3 mJ) compared with the main-exit-only method while lowering the accuracy from 69.9% to 66.2%.

Fig. 11(a)–(d) shows the power consumption and total energy cost to perform inferences. The processing time is

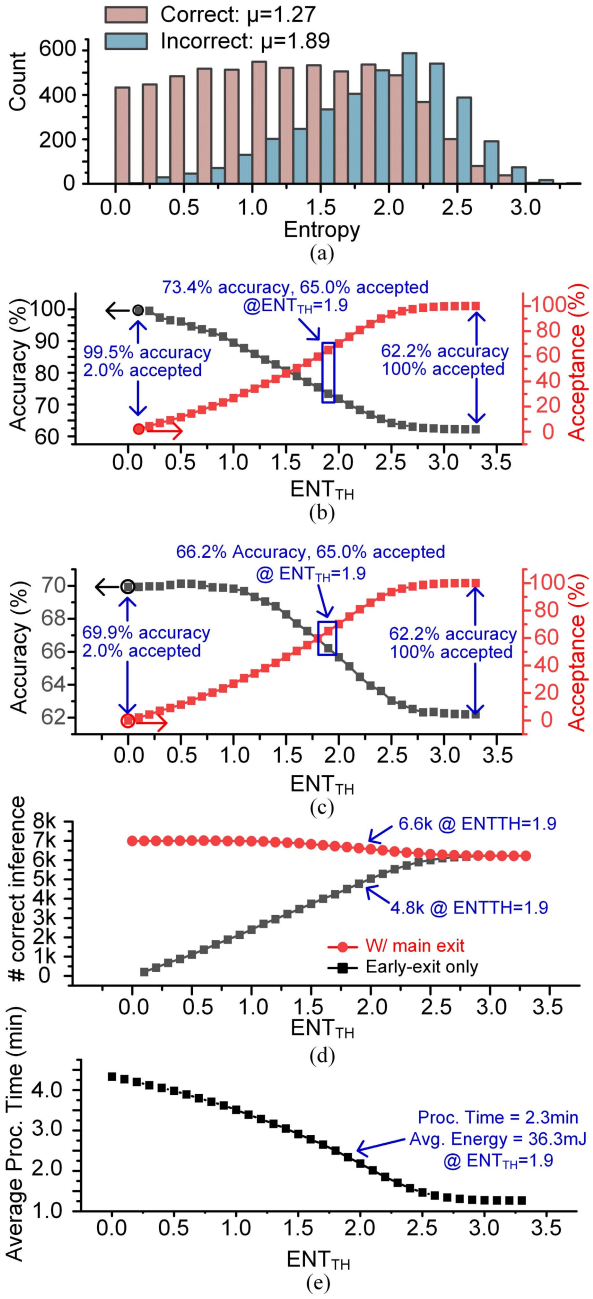

Fig. 10. Measured CNN for 10 000 images. (a) Entropy distribution. (b) Accuracy and acceptance ratio of only the early-exit path. (c) Accuracy acceptance ratio of the early-exit path followed by the main-exit path if entropy > ENT_{TH}. (d) Number of correct inferences. (e) Average processing time.

adjusted by the supply voltage and internal oscillator configuration. Fig. 11(e) and (f) shows their energy-delay product (EDP). At a supply voltage of 0.7 V and a clock frequency of 3.2 MHz, it achieves near-optimal EDP for both the main- and early-exit paths. Here, the early-exit path has a 3.4× shorter processing time and thus 3.4× lower energy consumption per inference than the main-exit path.

Fig. 12(a) shows the power consumption breakdown for four different functions. Compared with image sensing [41], the CNN inference consumes 233× higher power. Among the

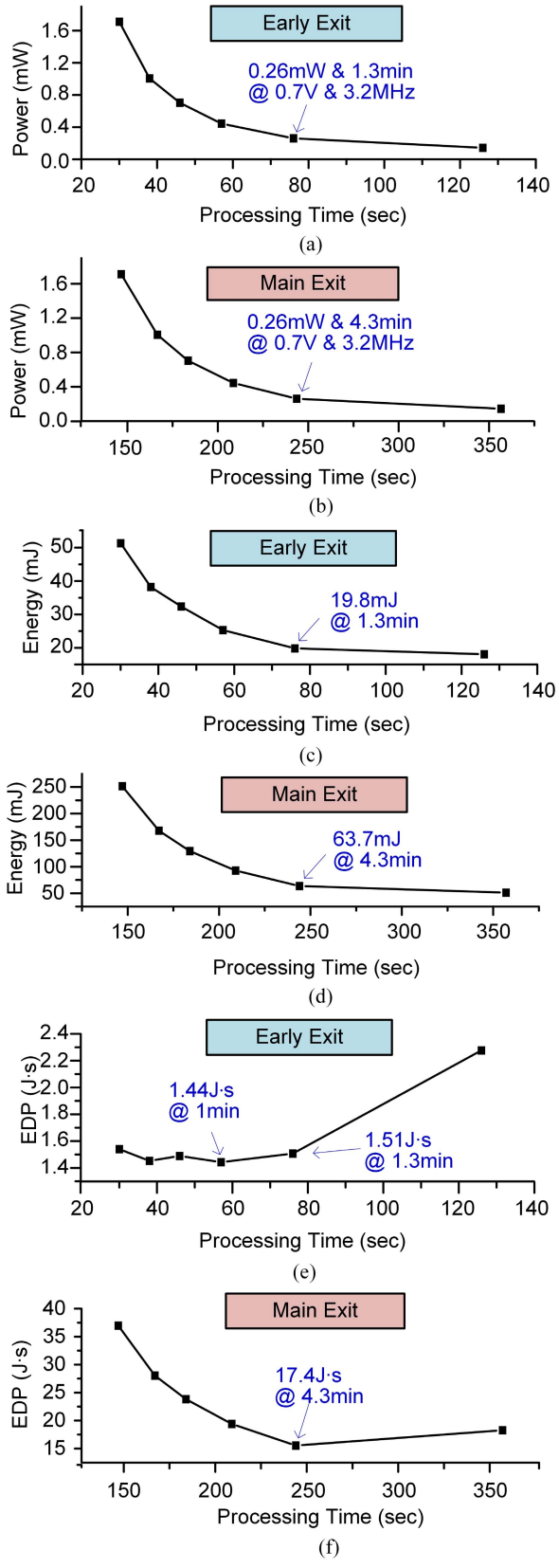

Fig. 11. Measured power and energy consumption across processing time. (a) Power consumption for the early-exit path. (b) Power consumption for the main-exit path. (c) Energy consumption for the early-exit path. (d) Energy consumption for the main-exit path. (e) EDP for the early-exit path. (f) EDP for the main-exit path.

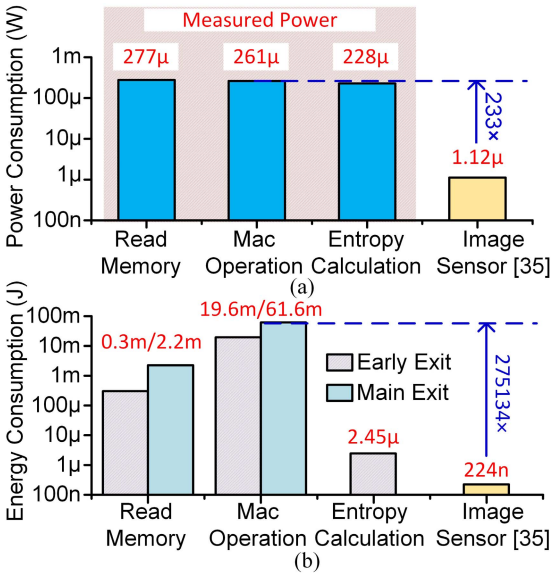

Fig. 12. Measured power and energy consumption of the CNN processing system compared with an image sensor. (a) Power consumption. (b) Energy consumption.

CNN operations, the reading memory and entropy calculation take up to 8 s, while MAC operations take 235 s. The early-exit path reduces the number of MAC operations from 130 00 000 to 500 000, finishes the inference faster by 3.1×, and saves energy by 68.8%.

In addition, we simulate the energy stored in a battery for a long-term operation and evaluate that the proposed CNN implementation can be sustained by a millimeter-scale energy harvester. The sunlight intensity is recorded every 5 min for more than 45 days (March 2020–April 2020) in the Beechwood Farms Nature Reserve in western Pennsylvania by five HOBO Pendant MX loggers (MX2202) in different places [42]. The harvesting power is measured across light intensities using a light energy harvester based on PV cells connected in series [37]. The simulated use scenario is that a system takes images every 30 min if the light level is stronger than 1 klx and infers the object in the image.

Assuming a battery with 1 J capacity, Fig. 13 shows the contour map of the average accuracy, required battery capacity, and average processing time, across $\mathrm{ENT_{TH}}$ and $\mathrm{BAT_{TH}}$. The accuracy varies from 62.8% to 67.9%, while the required battery capacity ranges from 120 to 4100 mJ. The average processing time is shorter than 4.4 min. Higher $\mathrm{BAT_{TH}}$ allows more early-exit inferences, while higher $\mathrm{ENT_{TH}}$ accepts more inference results of the early-exit path. For $\mathrm{ENT_{TH}}$ less than 1, the three parameters do not depend on $\mathrm{BAT_{TH}}$ since most inferences are completed by the main-exit path. For $\mathrm{ENT_{TH}}$ higher than 2, the parameters depend on both $\mathrm{BAT_{TH}}$ and $\mathrm{ENT_{TH}}$ since more early-exit inferences initiated by lower battery energy than $\mathrm{BAT_{TH}}$ becomes the final inference outputs. At the condition of either $\mathrm{ENT_{TH}} = 1.9$ or $\mathrm{BAT_{TH}} = 650$ mJ, the related $\mathrm{BAT_{TH}}$ and $\mathrm{ENT_{TH}}$ similarly affect the parameters.

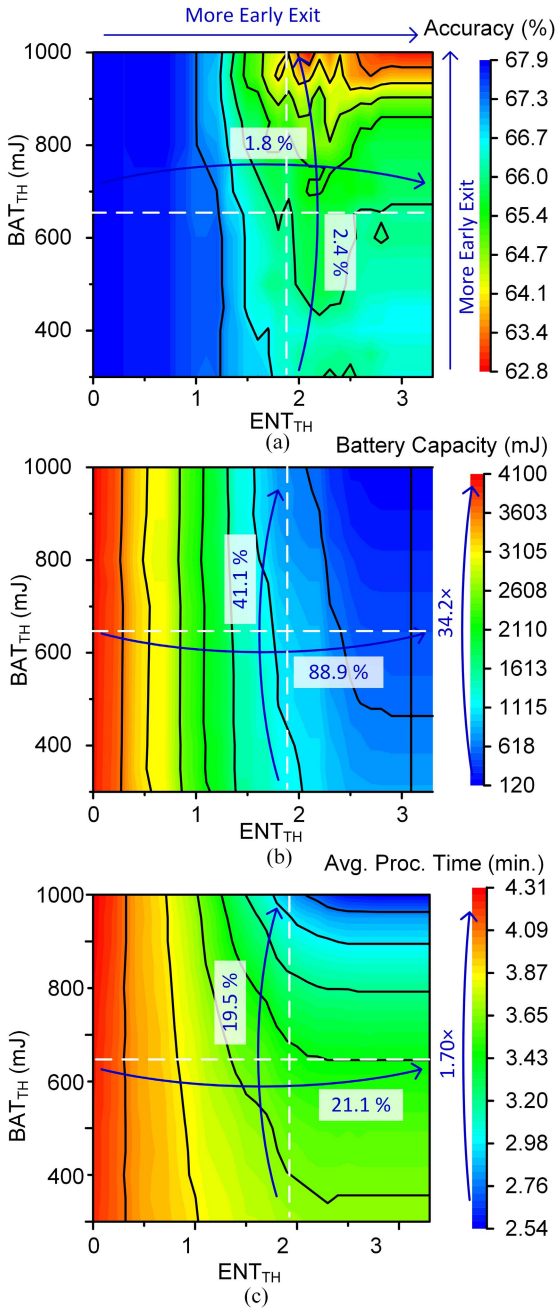

Fig. 13. Simulated accuracy, battery capacity, and average processing time across ENT_{TH} and BAT_{TH}. (a) Accuracy. (b) Battery capacity. (c) Average processing time.

Fig. 14 shows the remaining energy, for ENT_{TH} of 1.9 and BAT_{TH} of 700 mJ. It maintains the energy level stored in the battery under all the five environmental light intensity profiles, demonstrating energy-autonomous operation without battery replacement. The maximum energy drops of 855.3 mJ can be tolerated by including two millimeter-scale batteries. The targeted system includes two stacked batteries (684 mJ each), physically stacked in the die-stacking structure and electrically connected in parallel.

Fig. 15 shows the best accuracy at each processing time. For lower processing time, it requires both higher BAT_{TH} and

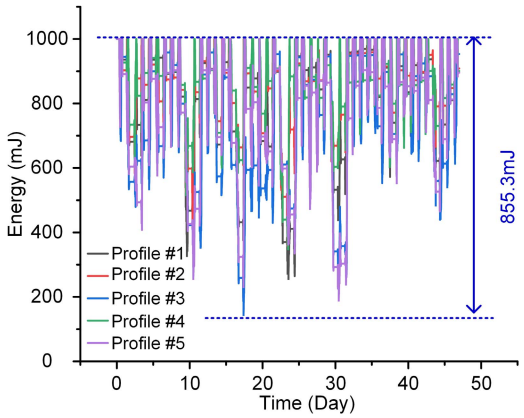

Fig. 14. Simulated remaining battery energy with 1-J battery capacity and different light intensities measured from five commercial light sensors.

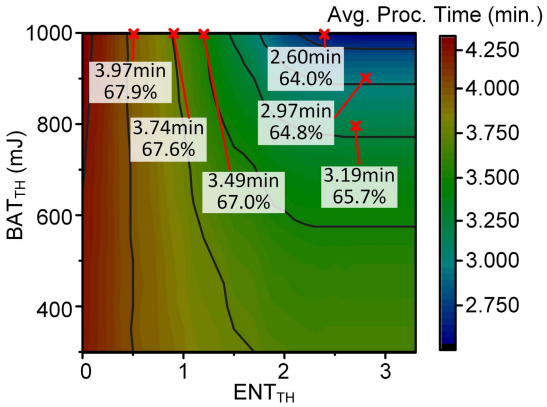

Fig. 15. Optimal ENT_{TH} and BAT_{TH} for the maximum accuracy at different averaging processing times.

ENT_{TH} to complete the inference computation at the early exit more and not use the main exit additionally. To reduce the processing time from 4.25 to 2.75 min (35% reduction), the inference accuracy needs to be sacrificed from 67.9% to 64.8% (4.6% reduction).

We also explore two different case studies with different requirements and investigate how the optimal ENT_{TH} and BAT_{TH} are changed for the different constraints. Fig. 16(a) considers a case where target minimum accuracy should be achieved. The design points, marked by "x," are the optimal ENT_{TH} and BAT_{TH} for different target minimum accuracies. The design point for 67.5% accuracy is not acceptable in the system since it requires more than two miniature batteries, which makes the system larger than the target. Fig. 16(b) considers the other case that maximizes the FoM for different weighting factors in the FoM. In addition to accuracy, it takes the battery capacity into account to decrease a custom battery size further and thus the system form factor. The FoM is defined as

$$\text{FoM} = \frac{(f(\text{accuracy}))^{\alpha}}{f(\text{minimum battery capacity})} \tag{3}$$

$$f(x) = 1 + \frac{x - \min}{\max - \min} \tag{4}$$

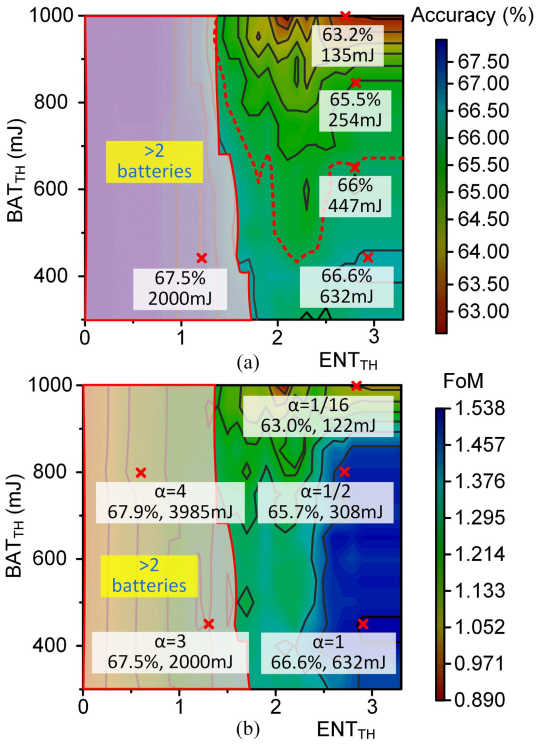

Fig. 16. Optimal ENT_{TH} and BAT_{TH} for different requirements. (a) Optimal points with the minimum battery capacity for different target accuracies. (b) Optimal points for best FoM with different weighting factors (α).

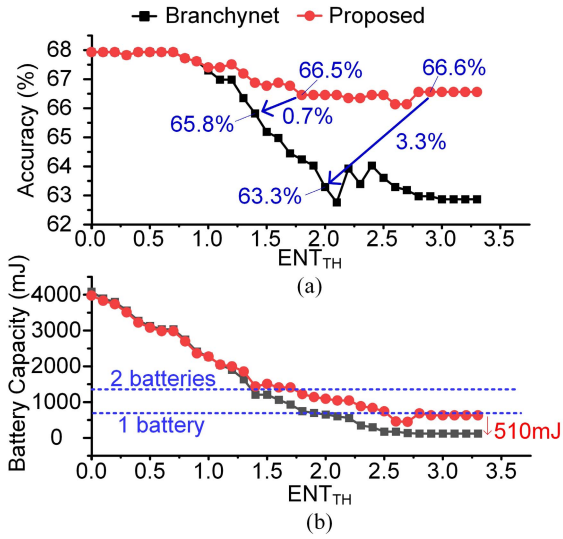

Fig. 17. Simulated proposed design and BranchyNet across ENT_{TH}. (a) Inference accuracy. (b) Required battery capacity.

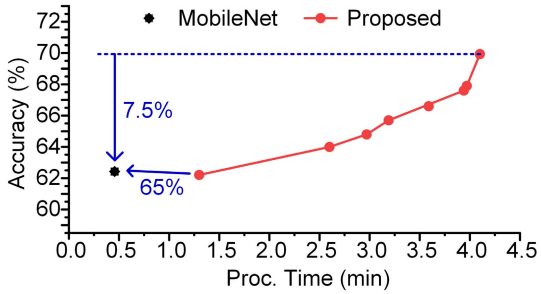

Fig. 18. Simulated inference processing time and accuracy of the proposed design and MobileNet.

where accuracy and battery capacity are mapped to a range between 1 and 2 and α is a weighting factor and controls the significance of accuracy compared to the required minimum battery capacity. For example, at α of 1, FoM becomes the highest at ENT_{TH} of 2.9 and BAT_{TH} of 450 mJ, obtaining an accuracy of 66.6% and requiring the minimum battery capacity of 632 mJ. Each marked point depicts the optimal ENT_{TH} and BAT_{TH} for the maximized FoM. The enclosed numbers are α, accuracy, and minimum battery capacity. The design points for α of 3 and 4 require two batteries or more, so they are not acceptable for the proposed system. As α increases, the optimal point moves to an area with higher accuracy, requiring larger battery capacity.

Fig. 17 compares the proposed design with another approach that allows early stopping of CNN processing for fast inference (BranchyNet) [20]. BranchyNet always processes the early-exit path and computes its entropy and moves to the main-exit path if the entropy is not low enough. In this simulation, BranchyNet is applied to the same CNN architecture of the proposed design. Compared with the proposed approach, this method does not consider the energy condition (e.g., BAT_{TH}) before processing the early-exit-only part. If the entropy is not low enough, energy and processing time for the early-exit-only part directly becomes a penalty. The proposed method avoids the risk at the cost of losing a chance to complete the computation only with the early-exit path with entropy lower than ENT_{TH}. In the CIFAR-10 dataset and the given light intensity profile, the two techniques achieve similar performance when ENT_{TH} is less than 0.9. For higher ENT_{TH}, the proposed design achieves higher accuracy up to 3.7% than BranchyNet at the cost of a larger required battery capacity up to 510 mJ. In the proposed miniature system, the battery capacity should be smaller than 1368 mJ. To meet this requirement, ENT_{TH} should be set to 1.8 for the proposed design and 1.4 for BranchyNet. Despite higher ENT_{TH} to complete more samples in the early-exit path, the proposed design achieves 0.7% higher accuracy. If the system is further miniaturized and able to include only a single battery (684 mJ), the advantage of the proposed design becomes more noticeable by achieving 3.3% higher accuracy.

Fig. 18 compares the proposed design with a state-of-the-art lightweight CNN architecture (MobileNet) [17], which can be applied to relatively small target CNN among many others. MobileNet employs the depthwise convolution that reduces the scale of the coefficients and thus the number of MAC operations. The depthwise convolution technique is applied to the target CNN architecture for comparison. In the CIFAR-10 dataset and the given light intensity profile, MobileNet dramatically reduces the processing time by 65% with similar accuracy to the early-exit path of the proposed design, but it cannot be accepted if a target accuracy is higher than 63%. Instead, in the proposed design, the accuracy can be

improved at the cost of processing time using two thresholds (BAT_{TH} and ENT_{TH}) between the early-exit or main-exit paths, which enables to satisfy different application requirements.

The proposed design provides the best performance with the optimally selected BAT_{TH} and ENT_{TH}. They need to be chosen by considering the environmental light intensity profile, required minimum accuracy, and available battery capacity. For a practical operation, the technique can be used with BAT_{TH} and ENT_{TH} with a margin from the optimal point to cover different environmental light intensity levels from the estimation. In the worst case, the batteries will be fully discharged, and the system will be shut down. However, the system restarts its operation again once the energy harvester recharges the battery enough, and data stored in the nonvolatile flash memory can be downloaded to a gateway.

VI. Conclusion

This article proposes an energy-aware adaptive NN implementation for a millimeter-scale sensing system. It includes two exit options for dynamic available energy conditions in a miniature system powered by an energy harvester. The implemented NN reduces processing time and thus energy consumption by 43.9%, compared with the main-exit-only approach while sacrificing its accuracy from 69.9% to 66.2%. Also, we explore the required minimum battery capacity at each optimal configuration for accuracy and/or energy consumption to achieve energy-autonomous operation under measured exemplary light profiles. It requires a minimum battery capacity of 855 mJ, acceptable for the target miniature system with two millimeter-scale batteries (684 mJ each). Compared with the state-of-the-art CNN technique (BranchyNet) allowing early stopping, the proposed design improves the accuracy by 0.7% and 3.3% to maintain energy-autonomous operation with two and one millimeter-scale batteries, respectively. Compared with the state-of-the-art lightweight CNN technique (MobileNet), this work provides flexibility with a tradeoff between accuracy and processing time for different application requirements.

Acknowledgment

The authors would like to thank the Audubon Society of Western Pennsylvania (ASWP), Pittsburgh, PA, USA, for their support in the data collection of environmental light profiles.

References

[1] Y. Lee *et al.*, "A modular 1 mm^3 die-stacked sensing platform with low power I^{2}C inter-die communication and multi-modal energy harvesting," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 229–243, Jan. 2013.

[2] V. Iyer, A. Najafi, J. James, S. Fuller, and S. Gollakota, "Wireless steerable vision for live insects and insect-scale robots," *Sci. Robot.*, vol. 5, no. 44, pp. 1–12, Jul. 2020.

[3] Y. Chen *et al.*, "An injectable 64 nW ECG mixed-signal SoC in 65 nm for arrhythmia monitoring," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 375–390, Jan. 2015.

[4] G. Kim *et al.*, "A millimeter-scale wireless imaging system with continuous motion detection and energy harvesting," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2014, pp. 1–2.

[5] A. H. Alavi *et al.*, "Self-charging and self-monitoring smart civil infrastructure systems: Current practice and future trends," *Proc. SPIE*, vol. 10970, Mar. 2019, Art. no. 109700W.

[6] *The CIFAR-10 Dataset*. Accessed: Apr. 5, 2022. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[7] Q. Dong *et al.*, "A 1 Mb embedded NOR flash memory with 39μW program power for mm-scale high-temperature sensor nodes," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 198–200.

[8] M. Diana, J. Chikama, M. Amagasaki, M. Iida, and M. Kuga, "Characteristic similarity using classical CNN model," in *Proc. 34th Int. Tech. Conf. Circuits/Systems, Comput. Commun. (ITC-CSCC)*, Jun. 2019, pp. 1–2.

[9] *Rechargeable Solid State Bare Die Batteries*, EnerChip Bare Die CBC005 Datasheet, CYMBET Corporation, Elk River, MN, USA, 2016.

[10] *Mixed Signal Microcontroller*, MSP430G2x11 MSP430G2x01 Datasheet, Texas Instruments, Dallas, TX, USA, 2013.

[11] I. Lee *et al.*, "A 179-lux energy-autonomous fully-encapsulated 17-mm^3 sensor node with initial charge delay circuit for battery protection," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 251–252.

[12] W. Jung *et al.*, "A 3 nW fully integrated energy harvester based on self-oscillating switched-capacitor DC-DC converter," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 398–400.

[13] I. Lee *et al.*, "A >78%-efficient light harvester over 100-to100 klux with reconfigurable PV-cell network and MPPT circuit," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2016, pp. 370–372.

[14] X. Wang, M. Magno, L. Cavigelli, and L. Benini, "FANN-on-MCU: An open-source toolkit for energy-efficient neural network inference at the edge of the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4403–4417, May 2020.

[15] L. Buonanno, D. Di Vita, M. Carminati, and C. Fiorini, "A directional gamma-ray spectrometer with microcontroller-embedded machine learning," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 4, pp. 433–443, Dec. 2020.

[16] Y. Li *et al.*, "Implementation of multi-exit neural-network inferences for an image-based sensing system with energy harvesting," *J. Low Power Electron. Appl.*, vol. 11, no. 3, p. 34, Sep. 2021.

[17] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[18] Y. Wu, Z. Wang, Z. Jia, Y. Shi, and J. Hu, "Intermittent inference with nonuniformly compressed multi-exit neural network for energy harvesting powered devices," in *Proc. 57th ACM/IEEE Design Automat. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[19] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence beyond the edge: Inference on intermittent embedded systems," in *Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, New York, NY, USA, Apr. 2019, pp. 199–213.

[20] S. Teerapittayanon, B. McDanel, and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 4–8.

[21] G. Huang *et al.*, "Multi-scale dense convolutional networks for efficient prediction," 2017, *arXiv:1703.09844*.

[22] Y. Li, Y. Wu, X. Zhang, E. Hamed, J. Hu, and I. Lee, "Developing a miniature energy-harvesting-powered edge device with multi-exit neural network," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[23] D. Fabbri, E. Berthet-Bondet, D. Masotti, A. Costanzo, D. Dardari, and A. Romani, "Long range battery-less UHF-RFID platform for sensor applications," in *Proc. IEEE Int. Conf. RFID Technol. Appl. (RFID-TA)*, Sep. 2019, pp. 80–85.

[24] S. Ma, N. Pournoori, L. Sydanheimo, L. Ukkonen, T. Bjorninen, and A. Georgiadis, "A batteryless semi-passive RFID sensor platform," in *Proc. IEEE Int. Conf. RFID Technol. Appl. (RFID-TA)*, Sep. 2019, pp. 171–173.

[25] V. Iyer, R. Nandakumar, A. Wang, S. B. Fuller, and S. Gollakota, "Living IoT: A flying wireless platform on live insects," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2019, pp. 1–15.

[26] S. J. Thomas, R. R. Harrison, A. Leonardo, and M. S. Reynolds, "A battery-free multichannel digital neural/EMG telemetry system for flying insects," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 5, pp. 424–436, Oct. 2012.

[27] S. Jeong, Y. Kim, G. Kim, and D. Blaauw, "A pressure sensing system with $\pm$0.75 mmHg (3σ) inaccuracy for battery-powered low power IoT applications," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2020, pp. 1–2.

[28] S. Oh *et al.*, "A 2.5 nJ duty-cycled bridge-to-digital converter integrated in a 13 mm^3 pressure-sensing system," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 328–330.

[29] I. Lee, E. Moon, Y. Kim, J. Phillips, and D. Blaauw, "A 10 mm^3 light-dose sensing IoT^2 system with 35-to-339nW 10-To-300klx light-dose-to-digital converter," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C180–C181.

[30] T. Kang *et al.*, "A 1.74.12 mm^3 fully integrated pH sensor for implantable applications using differential sensing and drift-compensation," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C310–C311.

[31] M. Cho *et al.*, "A 142 nW voice and acoustic activity detection chip for mm-scale sensor nodes using time-interleaved mixer-based frequency scanning," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 278–280.

[32] M. Cho *et al.*, "A 6×5×4 mm^3 general purpose audio sensor node with a 4.7μW audio processing IC," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C312–C313.

[33] K. D. Choo *et al.*, "Energy-efficient motion-triggered IoT CMOS image sensor with capacitor array-assisted charge-injection SAR ADC," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 2921–2931, Nov. 2019.

[34] P. Pannuto *et al.*, "MBus: An ultra-low power interconnect bus for next generation nanopower systems," in *Proc. ACM/IEEE 42nd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2015, pp. 629–641.

[35] L. Chuo *et al.*, "A 915 MHz asymmetric radio using Q-enhanced amplifier for a fully integrated 3×3×3 mm^3 wireless sensor node with 20 m non-line-of-sight communication," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 132–134.

[36] W. Jung, D. Sylvester, and D. Blaauw, "A rational-conversion-ratio switched-capacitor DC-DC converter using negative-output feedback," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2016, pp. 218–220.

[37] E. Moon, I. Lee, D. Blaauw, and J. D. Phillips, "High-efficiency photovoltaic modules on a chip for millimeter-scale energy harvesting," *Prog. Photovolt., Res. Appl.*, vol. 27, no. 6, pp. 540–546, Jun. 2019.

[38] A. Morel *et al.*, "Self-tunable phase-shifted SECE piezoelectric energy-harvesting IC with a 30 nW MPPT achieving 446% energy-bandwidth improvement and 94% efficiency," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 488–490.

[39] P. Xu, D. Flandre, and D. Bol, "Analysis, modeling, and design of a 2.45-GHz RF energy harvester for SWIPT IoT smart sensors," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2717–2729, Oct. 2019.

[40] Y. D. Kim *et al.*, "A 7 nm high-performance and energy-efficient mobile application processor with tri-cluster CPUs and a sparsity-aware NPU," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 48–50.

[41] I. Park *et al.*, "A 640×640 fully dynamic CMOS image sensor for always-on object recognition," *IEEE J. Solid-State Circuits*, vol. 55, no. 4, pp. 898–907, Apr. 2020.

[42] Y. Li *et al.*, "Feasibility of harvesting solar energy for self-powered environmental wireless sensor nodes," *Electronics*, vol. 9, no. 12, p. 2058, Dec. 2020.