An Efficient Analog Convolutional Neural Network Hardware Accelerator Enabled by a Novel Memoryless Architecture for Insect-Sized Robots

Iman Dadras
Intelligent Materials and Systems Laboratory (IMS Laboratory)
Institute of Technology
University of Tartu
Tartu, Eatonia
iman.dadras@ut.ee

Mohammad Hasan Ahmadilivani
Centre for Dependable Computing Systems
Department of Computer Systems
Tallinn University of Technology
Tallinn, Eatonia
mohammad.ahmadilivani@taltech.ee

Saoni Banerji
Intelligent Materials and Systems Laboratory (IMS Laboratory)
Institute of Technology
University of Tartu
Tartu, Eatonia
saoni.banerji@ut.ee

Jaan Raik
Centre for Dependable Computing Systems
Department of Computer Systems
Tallinn University of Technology
Tallinn, Eatonia
jaan.raik@ttu.ee

Alvo Abloo
Intelligent Materials and Systems Laboratory (IMS Laboratory)
Institute of Technology
University of Tartu
Tartu, Eatonia
alvo.aabloo@ut.ee

Abstract*—For decades, miniaturization of robots has gained considerable attention due to the exciting applications of insect-sized robots, such as ambient monitoring. However, scaling down the robots' dimensions reduces energy availability drastically for sensors and controllers. It has prohibited many successful technologies tested in larger-scale robots from application in insect-sized ones. As a result, insect-sized robots' power and sensor/control autonomy is an open field of research. One of these technologies is Convolutional Neural Networks (CNN). This paper presents novelty in different levels of abstraction from architectural to transistor-level that drastically reduces the CNN power to comply with the low power budget of insect-sized robots. Analog computation is utilized for its compactness, and an architecture is devised to simplify the analog circuitry. Proposed convolutional filters, showing four orders of magnitude higher efficiency with respect to the state-of-the-art, consume merely 1.5 nW/image with 92% accuracy and promise application of CNN-based controllers in insect-sized robots.

Keywords

analog CNN, hardware accelerator, memoryless CNN, mixed-signal design, low-power ASIC design, insect-sized robots

I. Introduction

Visual perception constitutes 90% of human brain input [1], and machine vision is proven to be a disruptive technology in robotics [2]. Convolutional Neural Networks (CNN) are used as a solution to perform machine vision tasks adapted from Artificial Intelligence (AI) domain for image classification problems [3], [4]. It is utilized in robots' locomotion control for applications such as obstacle avoidance [5], target detection [6], foothold selection [7], and trajectory planning [8]. However, CNN-based onboard locomotion control has only been deployed for relatively large-scale robots [9] owing to the high-power and area requirements of CNN processors and the low payload capacity of insect-sized robots [10], [11]. This poses a pressing need to reshape CNN processors for insect-sized robots in terms of size, weight, and power (SWaP) cost.

Recent work [9] has demonstrated the utility of CNN for visual control at insect-scale. A custom-built low-weight vision sensor is mounted on a flapping wing insect-sized robot. The images are classified using CNN implementation off-board to make the robot recognize and repeatably move toward flower images and away from predator images. However, owing to the computationally expensive [12] CNN algorithms and the payload and power constraints of the robot, the system is unable to accommodate onboard computation, restricting its field of operation.

Small payload capacity in the order of few hundreds of milligrams in insect-sized drones [10] and tens of milligrams in ionic electroactive polymer (IEAP)-based robots [11] is prohibitive for power and control autonomy. An insect-sized robot with extended payload capacity [10] is shown to have enough payload for either power or sensor autonomy, not both.

By shrinking the processor and reducing the power, Application-Specific Integrated Circuit (ASIC) hardware accelerators can improve both control and power autonomy in compliance with small robots' low power budget (100 µW to 100 mW for the whole system [13]). An autonomous 10-cm glider (MicroGlider) is demonstrated in [14]. MicroGlider has an audio-based guidance system assisted by an optic flow ASIC processor. BrainSoC [15] is a central controller designed for controlling insect-scale flapping-wing robots. It uses hardware accelerators for edge sharpening and optical flow. In [13], a Binary Neural Network (BNN) hardware accelerator is reported to have potential application in insect-sized drones.

To the authors' knowledge, no CNN hardware accelerator has been reported for onboard control of insect-sized robots.

This work was supported by H2020 project TWINNIMS (Grant agreement 857263), Estonian Research Council Grant 1084 and Estonian Centre of Excellence in ICT Research.

References [16]–[18] report low power integrations of CNN/BNN first layers with CMOS Image Sensors (CIS) for always-on devices. Although a camera and a CNN's first layer on the same chip reduces the energy-hungry inter-chip data transfer, a higher level of integration is required to comply with the restricted requirements of insect-scale robots. Reference [19] shows a CIS integrated with a full analog convolutional processor for an always-on image sensor. However, although this reference exploits analog computation compactness, the use of capacitors as memory components deteriorates its performance and hinders repurposing it for insect-sized robots.

Analog computation represents each pixel with a single signal (voltage or current) and performs multiply-accumulate (MAC) operations with approximately one transistor per input bit. This feature empowers analog computation compared to its digital counterpart, which needs several gates for each operation. Despite this advantage, interlayer memories in existing CNN architectures impede wide-spread utilization of analog processors. In a feed-forward CNN, the input of each layer is the output of the previous layer. Thus, the output features of a layer should await the completion of the rest of the features' map in a memory. This leads to massive memory walls between layers. Analog designers can realize these architectures either by several power-hungry conversions between analog and digital domains to store features in digital interlayer memories or by implementing slow and error-prone analog memories [19], [20], of which neither is appropriate. Pipeline architecture [21] has been proposed to minimize the memory walls. However, if the output is analog, it still needs to be either converted into digital domain to be stored in the memory, or designers are required to tackle the hassles of analog memory [22].

As the problem arises at the architecture level, a solution should be sought by an interlevel design in which algorithm and architecture are selected or designed for analog computation. In this work, with a hybrid bottom-up and top-down design approach, a new architecture is tailored for an optimal analog computational performance (bottom-up) in which memory is omitted completely. Then, an algorithm is selected to keep the architecture reasonably sized (bottom-up) by avoiding overlaps between receptive fields of output map features while achieving the application requirements (top-down).

The proposed architecture is completely memoryless, i.e., no storing components such as capacitors are used as they slow down the system by introducing time constants to the circuit as well as increasing processing time and energy consumption per image. Therefore, designs such as [19] that do not have a memory block but use capacitors to store data after each clock cycle are not considered to be memoryless.

The convolutional processor in this paper, based on the new memoryless architecture and novel low-power analog circuitries (<1.5 nW/image), fits the low power requirement and complies with the restricted power budget of insect-sized robots. Our contributions towards the first autonomous insect-sized robot that will have a single-chip control unit including a full CNN inference engine, controller, and CIS is as follows:

- We proposed a new architecture termed Funnel that omits the need for intermediate memories and ADC/DACs and lessens the requirements on output ADC significantly.

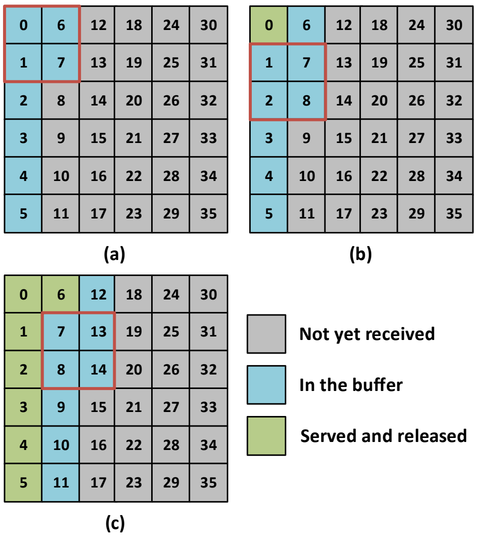

Fig. 1 Pipeline architecture (a) enough pixels in the buffer for one kernel operation (red box). (b) kernel operation is done, pixel number 0 is released from memory to free space for pixel number 8. (c) the process has progressed for six clocks. (Taken from [21])

- Novel circuitries are proposed to realize the Funnel architecture, among them a dual-purpose input DAC/convolution circuitry which performs both operations with about one transistor per input bit.

- It is shown that the proposed analog convolutional processor is in four orders of magnitude more efficient than [19], achieving 46 TOPSPW without sacrificing accuracy.

The rest of the paper is organized as follows: Section II presents the novel convolutional processor architecture. Then, circuit topology for each layer is proposed in Section III. Section IV explains the experimental setup. The interpretation of the results is discussed in Section V, followed by concluding remarks in Section VI.

II. Proposed Funnel Architecture

With contemporary computer architecture, interlayer memory walls are responsible for the significant area and power dissipation in CNN accelerators [23]. Pipeline architecture was devised to shrink the memory walls between the intermediate layers [21]. However, it fails to ease the ADC/DAC requirements. For an ADC/DAC-free accelerator, the circuit should completely lose the intermediate memories.

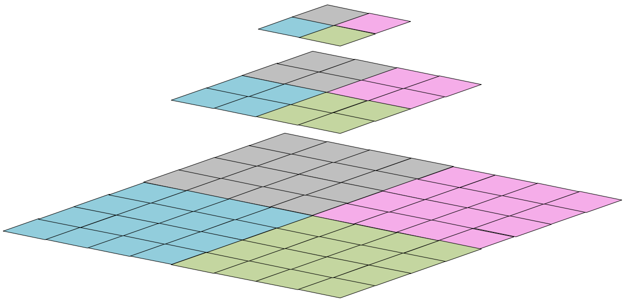

Fig. 2 Proposed Funnel architecture. Bottom: input layer. Center: intermediate layer. Top: output layer.

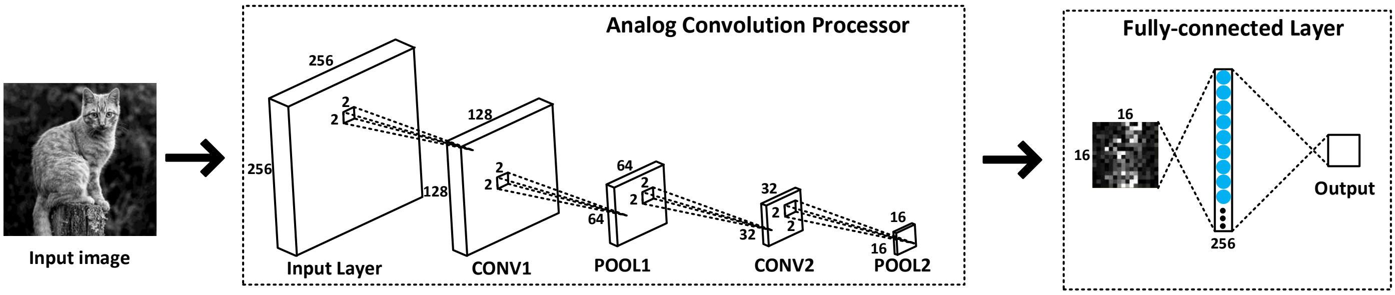

Fig. 3 LWCNN algorithm adapted from [19]

In the pipeline architecture, the features are stored in a buffer where they are awaiting the rest of the array. When there are sufficient features for a kernel operation in the next layer, the accelerator conducts the operation. Then, the accelerator discards any used features that are not involved in forthcoming operations to free up space for new features from the previous layer (Fig. 1).

The architecture proposed in this paper is based on two additional aspects to pipeline architecture. First, kernel operations are spatial. It means that the order of pixels in input register matters. For example, in Fig. 1, if the order of incoming pixels changes to 0, 1, 6, and 7, the number of required buffer registers is reduced to 4. Moreover, when only 4 pixels need to be accessed simultaneously, they could be obtained concurrently with parallel computation instead of using memory.

Therefore, the first layer throughput is selected equal to the receptive field of the first feature in the convolutional processor output. Then, the accelerator conducts parallel computations and provides the intermediate layers with the required input to produce one feature at the final output. The throughput then moves to the receptive field of the second output feature and so forth.

Fig. 2 shows funnel architecture for a small convolutional network with one intermediate layer and stride and kernel size 2×2. As only one output is produced at each clock, the requirements on the output ADC are also reduced.

A. LWCNN Algorithm

This paper adapts the LightWeight Convolutional Neural Network (LWCNN) algorithm [19] according to the applied image resolution due to the reasons stated below:

- The funnel architecture has the best efficiency for algorithms in which the stride and kernel size is equal in each layer. In this condition, the receptive fields of output features share no pixel. If receptive fields share pixels, a bigger throughput is needed to prevent redundant calculations.
- The algorithm consists of only four layers and conforms to the SwaP cost for insect-sized robotics. The algorithm has been tested successfully [19].

The modified LWCNN algorithm is illustrated in Fig. 3. The convolutional processor consists of two convolutional and two pooling layers. The kernel in each layer is 2×2 with a stride of 2. The last pooling output is 256 features which are processed in the fully-connected layer for binary classification.

III. CNN Accelerator Design

Two different convolution circuits, two versions of a maximum pooling circuit, and a fully-connected unit are used to realize the LWCNN algorithm with the funnel architecture. The input circuitry performs both digital to analog conversion and convolution. The second layer is a modified voltage-mode MAX circuit. Layer three conducts convolution with a differential pair-like circuit. And the last layer is again a voltage-based MAX.

The Funnel architecture requires simultaneous readiness of input for each layer. Shift registers transfer input pixels to the convolution layer. Then, 64 convolution blocks in the first layer (CONV1) provide inputs for 16 first pooling layer blocks (POOL1). The output of 16 POOL1 blocks goes to 4 CONV2 blocks that feed the last layer, POOL2, which produces one output that is converted to digital and handed to the digital fully-connected layer. The top-level topology is depicted in Fig. 4.

A. Dual-Purpose DAC/Convolution Input Circuitry

As mentioned, the power consumption of ADC/DAC blocks is the primary bottleneck in realization of CNN analog accelerators. Whereas the new funnel architecture omits the intermediate ADC/DAC and reduces the output ADC requirement to one pixel per clock, the proposed input layer precludes the input DAC as the conversion is carried out concurrently with the convolution.

The input layer circuitry is designed considering that DAC and convolution functionally resemble each other. For DAC, each bit is multiplied by its weight; then, all the products are summed up together according to (1).

$$P = \sum_{i=0}^{N-1} b_i W_{bi} \qquad (1)$$

In (1), b_i is the bit value, W_{bi} is the bit weight, N is the number of bits used to represent each pixel, and P is the pixel value.

Similarly, convolution result is the summation of all pixel-weights' products as shown in (2):

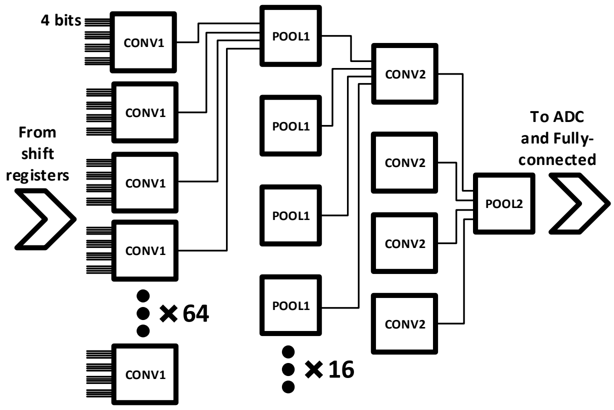

Fig. 4 Top level topology of analog convolutional network

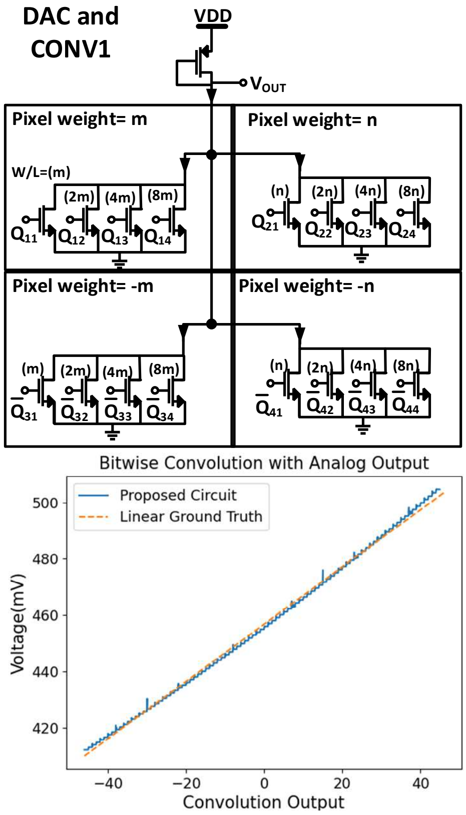

Fig. 5 Transistor-level schematic of the input layer circuitry and its output voltages for all possible convolution values

$$C = \sum_{i=0}^{K-1} P_i W_{pi} \qquad (2)$$

where C is the convolution output, W_p is the pixel weight, and K is the number of pixels in a kernel.

By factorization, it is possible to multiply each bit with the product of bit and pixel weights and then add all the results together, directly obtaining convolution output:

$$C = \sum_{i=0}^{(N-1)(K-1)} b_i W_{ni} \qquad (3)$$

W_n is $W_b \times W_p$.

The first convolution block is implemented by the use of current sources controlled by bit values. For each bit, a current source transistor, e.g., a simple common-source NFET, is placed. The aspect ratios of current sources are set proportional to W_n of the corresponding bit. The current sources for bits with negative and positive pixel weights are normally-on (gates are connected to the $\overline{\mathrm{Q}}$ output of the last flip-flop of input image shift register) and off (connected to Q output), respectively. Drain currents of all transistors go to the load transistor generating a proportional voltage. Hence, when all bits are zero, there is a neutral voltage (corresponding to zero) at the output. As any bit turns one, its current source turns on for bits in pixels with positive weight and off for bits in pixels with negative weights So, the output voltage changes proportional to W_n . Fig. 5 shows the transistor-level implementation of the convolution block with a kernel size of four 4-bit input pixels. The first and second digits in the Q subscript indicate the pixel number the bit number, respectively. This figure also illustrates the convolution block output. With four 4-bit bytes with weights of [-2, -1, 1, 2] the output can have 92 values. The results are almost linear. It is worth mentioning that in case of a Relu activation function, the negative values are disregarded, leading to an even more linear output. According to the architecture and the algorithm, 64 instances of this block are needed in the input layer. Thus, this part highly influences the power and area of the circuit.

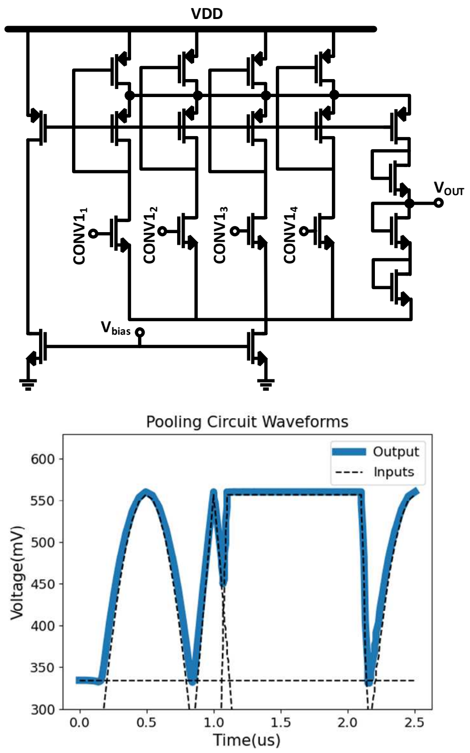

Fig. 6 The modified voltage-mode MAX circuit and standard voltage-mode MAX circuit input and output waveforms

B. Pooling Circuitry

Pooling (POOL1 and POOL2) blocks are modified versions of the widely used voltage-mode MAX pooling circuit [19] (see Fig. 6). However, as in this paper, the first pooling output should be assigned directly to a differential pair-like circuit; it is essential to have an adequate DC component for biasing the next stage and have small swinging to maintain the following circuit in the saturation region. Also, the impedance at the output node should match the one at the corresponding node of the input branches. Therefore, three transistors are placed at the output to add more degrees of freedom in the first pooling layer to meet these requirements (Fig. 6). The second pooling layer is a standard voltage-mode MAX circuit [24] and has only one NMOS in the output branch. The waveforms of the output pooling block are depicted in Fig. 6. The output follows the maximum input voltage performing pooling operation.

C. Second Convolution

A differential pair configuration with two parallel transistors on each side forms the CONV2 blocks. The schematic diagram is shown in Fig. 7. NFET aspect ratios and subsequently transconductances are set proportional to the weights. NMOS transistors corresponding to positive and negative weights are shown on the right and left-hand sides, in Fig. 7, respectively. Therefore, they add or subtract a proportionate current on the PMOS load transistor. Inputs/output waveforms are illustrated in Fig. 7. Output slope

TABLE I PERFORMANCE COMPARISON

	[16]	[17]	[18]	[27]	[19]	This work
Technology	Samsung 65 nm	180 nm	180 nm	45 nm	Dongbu 110 nm	TSMC 40 nm
Application	Face recognition	Classification	Classification	Feature extraction	Face detection	Face detection
Implemented analog circuitry	1st layer of CNN	1st layer of CNN	1st layer of BNN	Kernel filter and ReLu function	2 layers of CONV and POOL	2 layers of CONV and POOL
Supply (v)	1.2	0.5	1.8	1	3.3	0.9
Network Accuracy	96.18%	92.2%	98.3%	-	89.33%	92.2%
Resolution	320×240	128×128	32×32	120×120	160×120	256×256
Power Consumption	10.17-18.75 μW	117 μW	5.9 μW	11.44 mW (per kernel)	1.12 mW	97 μW
Efficiency (TOPSPW)	5.18-9.06	9.08	8.23	0.1	0.002	46.73

*The shaded works did not report complete convolutional processors

in each instance is equal to the weighted sum of the slope of inputs.

D. *Fully-Connected*

The digital fully-connected layer is placed after the convolutional processor. This layer conducts weighted-sum operations on the analog part output to categorize images into two classes.

At each clock cycle, the convolutional processor via an ADC provides the fully connected layer with one feature. The feature is multiplied by its corresponding weights and added to the values of two registers attributed to each class. At the last clock of each image, the fully connected layer determines the image's class according to the registers' values and resets the registers. Further details of the implementation of fully-connected layer are beyond the scope of this work.

IV. EXPERIMENTAL SETUP

The transistor-level implementation of the proposed circuit is simulated in TSMC 40nm technology using Cadence Design Suite together with the Spectre Simulator. Shift registers for input images and fully-connected layer are implemented in VHDL and included in the design to evaluate the final performance and accuracy of the LWCNN with proposed analog convolutional circuitry for face detection.

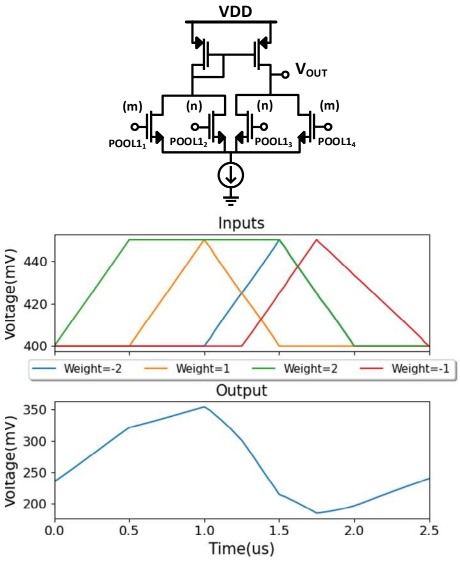

Fig. 7 Second convolution layer Schematic diagram and waveform

The training is carried out by a dedicated script written in Python. A dataset of 2700 gray-scale images (resolution 256 × 256) consisting of 1000 images of human faces (randomly taken from LFW dataset [25]) and 1700 images of cats and dogs (randomly selected from [26] is used. For power and memory considerations, all images, in both training and inference phases, weights, and the ADC are 4-bit.

The obtained weights are embedded into the VHDL of the fully-connected layer. The inference is conducted in Cadence Design Suite. Four hundred fifty images (150 humans, 300 animals from the aforementioned datasets) are analyzed to attain the system's accuracy.

V. RESULTS

The results of the convolutional network are assessed in two ways. First, the output of the proposed circuit for an image is compared with its ground truth obtained with a Python script. The comparison result shows good conformity: the simulated results and ground truth are visualized side by side in Fig. 8. It also shows the error distribution. The error for 188 pixels out of 256 is less than 5%. Mean square error (MSE) of the normalized values is calculated 0.007. The error does not affect the final accuracy of the system.

Then, according to the previous section, the convolutional network is simulated along with shift registers at the input and an ADC and fully-connected layer at the output to evaluate and compare the CNN system performance with cutting-edge accelerators. Table I compares the proposed accelerator with state-of-the-art CNN processors [16]–[19] and [27]. This table provides a thorough comparison of different works regarding their characteristics of the design (technology, implemented circuit, and supply voltage) as well as their applications. It also reports the accuracy of the neural networks with their image

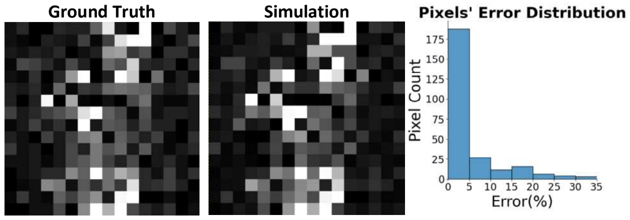

Fig. 8 Visualization of simulation results and ground truth with pixels' error distribution

resolutions and power and efficiency of implemented analog circuitry. The efficiency is measured using a criterion called Tera (MAC) Operation Per Second Per Watt (TOPSPW), which is the number of (MAC) operations is done in a processor normalized by power and time to give an unbiased comparison. Shaded columns represent partial analog implementations of the convolutional processor while this work and [19] realize a complete analog convolutional processor. References [16] and [18] consume less power than our work. However, they implement only one layer of network, and the efficiency criterion shows 5-9 times improvement, normalized by network size. In addition, the efficiency of our work is 23365 times, and the power consumption is 23.9 times better than [19] which used the same LWCNN algorithm.

VI. Conclusion

The paper presented a convolutional processor for CNN hardware acceleration based on a novel architecture. The proposed Funnel architecture omits the need for memories and dedicated ADC/DAC stages within the convolutional processor. A dual-purpose input layer for the convolutional processor was designed to satiate the accelerator with DAC while also performing convolution with almost a single transistor per an input bit. The circuit performance was compared to that of existing accelerators and showed competitive performance with significantly less power consumption (<1.5 nW per image) than provided by the state-of-the-art. The achieved computational efficiency in terms of TOPSPW was four orders of magnitude (more than 20,000 times) higher than the previous implementation of LWCNN algorithm. This extremely low power consumption and efficiency are promising to empower insect-sized robots with machine learning and modern robotic solutions.

References

[1] D. Hyerle, A Field Guide to Using Visual Tools, 0 ed. Assn for Supervision & Curriculum, 2000.

[2] X. Sun, X. Zhu, P. Wang, and H. Chen, "A Review of Robot Control with Visual Servoing," in 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Jul. 2018, pp. 116–121, doi: 10.1109/CYBER.2018.8688060.

[3] D. Yu and L. Deng, "Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP," IEEE Signal Process. Mag., vol. 28, no. 1, Jan. 2011, doi: 10.1109/MSP.2010.939038.

[4] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proc. IEEE, vol. 105, no. 12, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.

[5] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, "Automatic obstacle avoidance of quadrotor UAV via CNN-based learning," Neurocomputing, vol. 402, pp. 346–358, Aug. 2020, doi: 10.1016/j.neucom.2020.04.020.

[6] W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian, and Y. Zheng, "Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot," Comput. Electron. Agric., vol. 172, p. 105380, May 2020, doi: 10.1016/j.compag.2020.105380.

[7] O. A. V. Magana et al., "Fast and Continuous Foothold Adaptation for Dynamic Locomotion Through CNNs," IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 2140–2147, Apr. 2019, doi: 10.1109/LRA.2019.2899434.

[8] Y. J. Choi, T. Rahim, I. N. A. Ramatryana, and S. Y. Shin, "Improved CNN-Based Path Planning for Stairs Climbing in Autonomous UAV with LiDAR Sensor," in 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jan. 2021, pp. 1–7, doi: 10.1109/ICEIC51217.2021.9369805.

[9] S. Balasubramanian, Y. M. Chukewad, J. M. James, G. L. Barrows, and S. B. Fuller, "An Insect-Sized Robot That Uses a Custom-Built Onboard Camera and a Neural Network to Classify and Respond to Visual Input," in 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), Aug. 2018, pp. 1297–1302, doi: 10.1109/BIOROB.2018.8488007.

[10] S. B. Fuller, "Four Wings: An Insect-Sized Aerial Robot With Steering Ability and Payload Capacity for Autonomy," IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 570–577, Apr. 2019, doi: 10.1109/LRA.2019.2891086.

[11] I. Dadras et al., "Modeling and Experimental Analysis of the Mass Loading Effect on Micro-Ionic Polymer Actuators Using Step Response Identification," J. Microelectromechanical Syst., vol. 30, no. 2, Apr. 2021, doi: 10.1109/JMEMS.2021.3060897.

[12] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," Neural Comput. Appl., vol. 32, no. 4, Feb. 2020, doi: 10.1007/s00521-018-3761-1.

[13] A. Di Mauro, F. Conti, P. D. Schiavone, D. Rossi, and L. Benini, "Always-On 674μ W@4GOP/s Error Resilient Binary Neural Networks With Aggressive SRAM Voltage Scaling on a 22-nm IoT End-Node," IEEE Trans. Circuits Syst. I Regul. Pap., vol. 67, no. 11, pp. 3905–3918, Nov. 2020, doi: 10.1109/TCSI.2020.3012576.

[14] R. J. Wood et al., "Design, fabrication and initial results of a 2g autonomous glider," in 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005., 2005, p. 8 pp., doi: 10.1109/IECON.2005.1569190.

[15] X. Zhang et al., "A multi-chip system optimized for insect-scale flapping-wing robots," Jun. 2015, doi: 10.1109/VLSIC.2015.7231246.

[16] J.-H. Kim, C. Kim, K. Kim, and H.-J. Yoo, "An Ultra-Low-Power Analog-Digital Hybrid CNN Face Recognition Processor Integrated with a CIS for Always-on Mobile Devices," in 2019 IEEE International Symposium on Circuits and Systems (ISCAS), May 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702698.

[17] T.-H. Hsu et al., "A 0.5-V Real-Time Computational CMOS Image Sensor With Programmable Kernel for Feature Extraction," IEEE J. Solid-State Circuits, vol. 56, no. 5, pp. 1588–1596, May 2021, doi: 10.1109/JSSC.2020.3034192.

[18] Z. Li, H. Xu, L. Luo, Q. Wei, and F. Qiao, "A 5.9μW Ultra-Low-Power Dual-Resolution CIS Chip of Sensing-with-Computing for Always-on Intelligent Visual Devices," in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), May 2021, pp. 1–5, doi: 10.1109/ISCAS51556.2021.9401338.

[19] J. Choi, S. Lee, Y. Son, and S. Y. Kim, "Design of an always-on image sensor using an analog lightweight convolutional neural network," Sensors (Switzerland), vol. 20, no. 11, pp. 1–14, May 2020, doi: 10.3390/s20113101.

[20] B. Rumberg, S. Clites, H. Abulaiha, A. DiLello, and D. Graham, "Continuous-Time Programming of Floating-Gate Transistors for Nonvolatile Analog Memory Arrays," J. Low Power Electron. Appl., vol. 11, no. 1, p. 4, Jan. 2021, doi: 10.3390/jlpea11010004.

[21] A. Shafiee et al., "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Jun. 2016, pp. 14–26, doi: 10.1109/ISCA.2016.12.

[22] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," J. Phys. D. Appl. Phys., vol. 51, no. 28, p. 283001, Jul. 2018, doi: 10.1088/1361-6463/aac8a5.

[23] J. Reuben, "Binary Addition in Resistance Switching Memory Array by Sensing Majority," Micromachines, vol. 11, no. 5, p. 496, May 2020, doi: 10.3390/mi11050496.

[24] M. Soleimani, A. Khoei, K. Hadidi, and S. K. Nia, "Design of high-speed high-precision voltage-mode MAX-MIN circuits with low area and low power consumption," in 2009 European Conference on Circuit Theory and Design, Aug. 2009, pp. 351–354, doi: 10.1109/ECCTD.2009.5274998.

[25] E. L.-M. Gray B. Huang, Manu Ramesh, Tamara berg, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.," Amherst, 2007.

[26] "Dogs and Cats Images." https://www.kaggle.com/chetankv/dogs-cats-images (accessed Jun. 29, 2021).

[27] U. De Silva, S. Mandal, A. Madanayake, J. Wei-Kocsis, and L. Belostotski, "RF-Rate Hybrid CNN Accelerator Based on Analog-CMOS and Xilinx RFSoC," in 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Oct. 2020, pp. 1–5, doi: 10.1109/ISCAS45731.2020.9180556.