Ultra-low energy pest detection for smart agriculture

Davide Brunelli*, Tommaso Polonelli†, Luca Benini†‡
*Department of Industrial Engineering - DII, University of Trento, Italy
davide.brunelli@unitn.it
†Department of Electrical, Electronic and Information Engineering - DEI, University of Bologna, Italy
{tommaso.polonelli2, luca.benini}@unibo.it
‡Integrated Systems Laboratory, ETH Zurich, Switzerland
lbenini@iis.ee.ethz.ch

***Abstract*—Apple is one of the most produced fruits in the world because it is easy to grow, store, and transport. The most significant threat of this crop is the attack of the codling moth, a small insect capable of damaging whole orchards in a few days. To prevent this parasite and to plan effective countermeasures, we present an ultra low power smart camera capable of detecting and recognizing the pest in the field; therefore, a wireless alarm can be transmitted over a long distance. The system implements a machine learning approach based on neural networks on the camera board. The sensor is also provided with long-range radio capability and an energy harvester; it permits to operate indefinitely because of its positive energy balance when deployed in the field. Experimental tests on the proposed energy-neutral smart camera demonstrate a validation accuracy of 93% and only 3.5mJ required for image analysis and classification.**

I. Introduction

In the last years, information and Communications technologies (ICT) that can optimize the harvest of fruit and vegetables have gained momentum. In particular, over-population and the rise of the climate crisis necessitates smart technologies capable of minimizing the waste of resources and protecting the cultivations [1], [2]. For example, biological invasions could have severe global consequences if not handled correctly, including ecological destruction and economic losses. Thus, if not tackled promptly, crop losses and pest control can be extremely expensive [3].

This paper presents an ultra low power smart camera to detect and identify dangerous insects in the field, to trigger an alarm to the farmer, and to minimize the damage caused by pests in the orchard. Currently, the technologies deployed in precision agriculture are energy-hungry [4], and allows only to send raw images to the farmer so that one can decide if and how to counteract [1]. The proposed system's innovation combines sophisticated low energy sensors for the acquisition step, an efficient algorithm for the on-sensor analysis based on machine learning, with long-range communication. The system is designed for minimum power consumption; thus, it can also be energy autonomous thanks to the use of a small-size energy harvesting circuit.

Adding intelligence to the nodes and shifting the decision of anomalies near the sensor permits faster decisions and actions that are the key to damage reduction. Moreover, the low cost of the platform, its non-intrusive size, and ultra-low power design permit the high scalability of this solution in vast orchards. The in-field sensing model developed in this project belongs to the newest generation of agricultural sensing and automation devices, which can monitor and send real-time messages.

II. System Architecture

The system consists of a trap that looks like a small hive as shown in Fig. 1, where a pheromone bait and a glue layer capture the attracted insects even at an early stage when their density in the field is low. The farmer usually takes periodic inspections of the traps or mounts a wireless camera that sends the captured pictures wirelessly for remote evaluation. This process is expensive and time consuming. The proposed work detects the presence of the parasites thanks to a machine learning approach that triggers notifications and their position to the farmer, only when threats are recognized.

Fig. 1. Prototype of the IoT neural network Codling Moth smart trap.

The smart camera board consists of an ultra-low power smart camera with on-situ reconfigurable AI-capabilities. A multi-core architecture named PULP[1], implemented on a chip

[1]PULP: https://pulp-platform.org/

called GAP8 realized for IoT-inspired applications, is used to process the images.

GAP8[2] [5] SoC is a RISC-V ISA multi-core processor and integrates a state-of-the-art microcontroller, with a powerful programmable parallel processing engine for flexible multi-sensor (image, audio, inertial) data analysis, and a rich set of peripherals. The battery lifetime of the applications can be maximized by a powerful on-chip power management. The microarchitecture is mainly composed of two blocks, the Fabric Controller (FC) and the Cluster. The first is an advanced MCU based on a RISC-V core. It features an extended ISA for energy-efficient digital signal processing equipped with a fast access-time data memory (L1). Moreover, it includes a full set of peripherals (i.e., CAMERA Interface, I2S, QSPI, I2C) enabling CMOS camera and sound acquisition, in parallel with a 4-channel PWM interface for motor control, useful in applications such as Industrial IoT.

A multi-channel DMA manages the transfers between the peripherals and the RAM. The L2 memory size is 512 kB and, together with a ROM and an external 64 MB Hyper-RAM memory (L3), stores the code and most of the volatile variables. Using a dedicated frequency and voltage domain, the cluster is turned on when applications need computation-intensive functions. It contains 8 RISC-V cores identical to the FC, allowing the SoC to execute the same code on either the fabric controller or the cluster. The 8-core cluster shares L1 data memory. The shared L1 can serve all memory requests from the cores with single-cycle access latency and a low average contention rate.

The cluster program cache is also shared to maximize efficiency in fetching data-parallel code. The chip contains an internal DC/DC directly connected to an external battery or energy harvesters, used to optimize the power efficiency that is the primary requirement in low power applications. It provides voltage in 1.0 to 1.2 V range when the circuit is active, but when the SoC is in sleep mode, the DC/DC converter is turned off, and the system uses linear drop-out (LDO) regulator to power the RTC (real-time clock) and the part of the L2 memory that allows retention. In a deep sleep state, the current consumption is to 3.6 μW with the RTC always on and 30 μW, assuming full L2 retention.

One of the most power-consuming tasks in smart sensors for agriculture is transferring data and images acquired by the camera over long distances. Wireless communication is an essential feature, but it is usually the bottleneck for the data throughput and the power budget. We tackled both the challenges by i) minimizing the output data using a neural network based detection algorithm, and ii) we maximized the communication flexibility using an unlicensed wireless long-range protocol such as LoRaWAN [6]. The smart camera board designed for the project is shown in Fig. 2.

III. Data analysis

The dataset generation started with a small set of row pictures, as shown in Fig. 3 (approximately 300). The dataset is divided into two classes: *codling moth* (Fig. 4 left) and *general insects* (Fig. 4 right). The smart camera makes use of a gray-scale QVGA CMOS sensor (HiMax HM01B0) with a resolution of 244$\times$324 pixels. Hence it needs 79 KB of RAM to be stored (L2 in GAP8). To decrease the neural network complexity and the average power consumption, each image is pre-processed by a stack of morphological operators, which determines if the acquired image contains new elements of interest.

First of all, a histogram equalization filter is applied to decrease the brightness difference between successive frames. It is usually implemented in C++ code [7] and needs floating-point operations. The optimized imported version running on GAP8 requires only 37 Cycles-per-Pixel (cpp) working on a fixed-point framework. On the original QVGA image the overall number of clock cycles is 2.9 M.

Afterward, a background subtraction algorithm is used to detect changes in image sequences and to highlight the information provided by the HiMax camera. Background subtraction is often utilized for detecting moving objects in images taken from static cameras, but in our case, it serves to identify new trapped insects. In this paper, we calibrated the OpenCV MOG function on our dataset, which is part of the Gaussian Mixture Model (GMM) foreground detection [8]. In-field results show that a history (α) of two images and the mixture depth (κ) of five are the best trade-off between complexity and accuracy in our specific application, therefore we used these settings for the algorithm on GAP8.

The memory impact of the GMM is not negligible. It needs enough space to store the foreground history, which is two QVGA images and its associated statistical parameters. In fact, each pixel must include all the representative parameters of a Gaussian, i.e., weight, mean, and variance, as floating-point

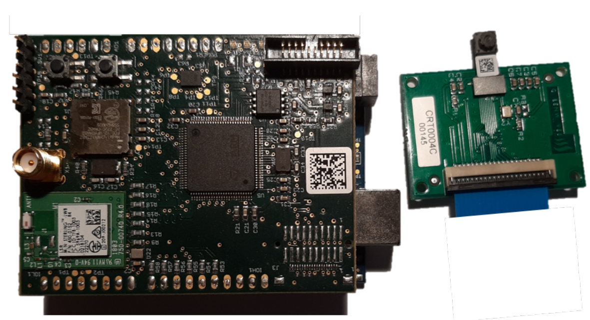

Fig. 2. Smart camera device used in the application.

[2]GAP8: https://greenwaves-technologies.com/gap8_gap9/

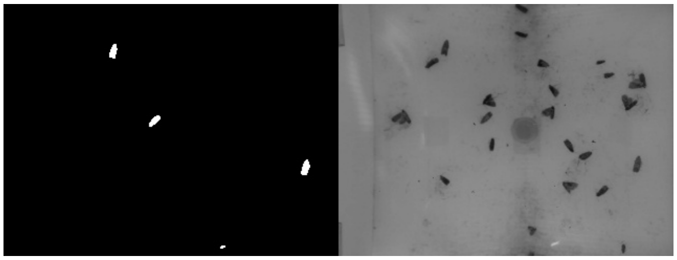

Fig. 3. Left: processed image, white insects was not presented in previous frames. Right: raw picture.

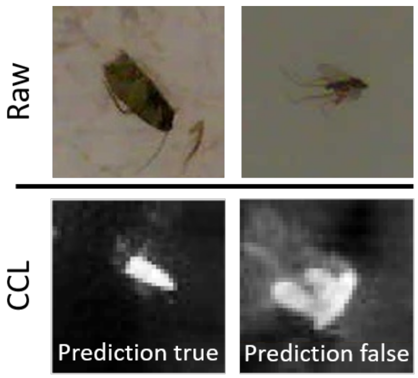

Fig. 4. Cropped Codling Moth (left). Cropped general insect (right) and their respective data analysis results.

variables. The size of the aforementioned structure, nominated as *Gauss*, is calculated as Eq. 1.

$$GMM_{float} = Gauss_{size} \cdot 244 \cdot 324 \cdot \alpha \cdot \kappa = 9.5\ MB. \quad (1)$$

We implemented a lightweight version of the standard OpenCV MOG function for GAP8, by developing the firmware using only fixed-point math. The code execution is performed using a 32-bit depth, with Q16.16 number format, while the *Gauss* structure is considered as an 8-bit variable with different fractional settings. Weights and mean are represented respectively as Q1.7 and Q8.0 while the variance is in Q6.2 format. With this method, the GMM requires only $GMM_{fixed} = 1.2\ MB$, 5× less than GMM_{float} and 392 cpp executed on the GAP8's FC. In comparison with the standard OpenCV MOG function [8], the output degradation is below 1%, error mainly composed by isolate pixels not recognized as part of the background. The execution of GMM is the most power-hungry, requiring 60% of the overall energy. This algorithm is sequentially running on the FC, and will be further parallelized in future works. The next step consists of a cascade of morphological operators to remove the noise after the GMM and cut the non-interesting objects, those too big or too small to be a general insect. Two iterations of Closing (morph close [8]) and one Opening (morph open [8]) operators make the processed image ready for the last processing task. They need an average of 27.5 cpp. The last task consists of

TABLE I
DATA ANALYSIS STACK

Step	M Cycles	Time [ms]	Energy [mJ]
Hist. Eq.	2.9	58	0.20
GMM	30.9	618	2.10
Closing	4.3	86	0.29
Opening	2.2	44	0.15
CCL	0.2	4	0.01
Total	40.5	810	2.8

Time calculated considering the FC running at 50 MHz

a connected component (CCL) function crop and save (in L3) only the sub-parts (if any) of the original images that are interesting for the application, producing images of 40x40 pixels. This last layer of the stack requires 2.8 cpp. Table I summarizes the execution time and workload for the whole stack, which requires 40.5 MCycles and 2.8 mJ.

IV. Neural Network

To correctly analyze 40×40 images, we used a machine learning approach; one of the best ways to classify insects from images [9], [10], with an accuracy above 80%. A convolutional neural network has been specifically used for this application, namely Smart Agriculture Neural Network (SANN). The SANN topology was inspired by CIFAR10 networks [11] and was reduced in size and complexity to minimize the bare image processing time using NEMO[3] and DORY[4]. The SANN is composed of four successive convolutional layers (CCN) and four fully-connected layers, in combination with batch normalization and ReLu activation functions. As reported in [9], among other configurations, the four layers CCN reaches the best trade-off between performance and complexity in detecting insects.

It is trained on the small available dataset featuring a 97% training accuracy and 93% of validation accuracy. These results are also confirmed after the quantization procedure (NEMO), where the weights and coefficients are represented in fixed-point 8-bit format. Tests on GAP8 assess the computational complexity to 2.9 MCycles, considering the execution on the 8-cores cluster. The equivalent energy reaches 0.66 mJ; hence the overall energy consumption amounted to 3.5 mJ.

Finally, if a parasite is detected, GAP8 encodes the original images in a JPEG format [12] to decrease the data size, and successively, sends it remotely through LoRaWAN. Each image transmission requires up to 52 J.

By comparison, the average energy per bit of LoRaWAN is 2.2 mJ [13], [14], which is equivalent to the whole data

[3]NEMO: https://github.com/pulp-platform/nemo
[4]DORY: https://github.com/pulp-platform/dory

analysis in Section III. Hence, in this application, the edge-computing paradigm can save up to five orders of magnitude the energy required by the smart camera, avoiding wireless communications when they are not strictly necessary, i.e., when a codling moth is not detected.

V. Conclusions

This paper presents *mJ-class* smart camera with an embedded tiny machine learning model trained for precision agriculture services. The camera identifies the pests of apple in the orchards and triggers an alarm to the farmer. The extremely low power consumption permits the smart-trap to operate without any maintenance for years, using a LPWAN to transmit reports to several kilometers promptly. The proposed framework is likewise valid for other precision agriculture applications, by retraining the SANN with a deployment-specific database.

Acknowledgement

This work was supported in part by the Italian Ministry for Education, University and Research (MIUR) under the program "Dipartimenti di Eccellenza (2018-2022)".

References

[1] N. Gondchawar, R. Kawitkar *et al.*, "Iot based smart agriculture," *International Journal of advanced research in Computer and Communication Engineering*, vol. 5, no. 6, pp. 838–842, 2016.

[2] D. Sartori and D. Brunelli, "A smart sensor for precision agriculture powered by microbial fuel cells," in *2016 IEEE Sensors Applications Symposium (SAS)*, 2016, pp. 1–6.

[3] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Comput. Electron. Agric.*, vol. 123, no. C, pp. 17–28, Apr. 2016.

[4] D. Brunelli, A. Albanese, D. d'Acunto, and M. Nardello, "Energy neutral machine learning based iot device for pest detection in precision agriculture," *IEEE Internet of Things Magazine*, vol. 2, no. 4, pp. 10–13, 2019.

[5] E. Flamand, D. Rossi, F. Conti, I. Loi, A. Pullini, F. Rotenberg, and L. Benini, "Gap-8: A risc-v soc for ai at the edge of the iot," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2018, pp. 1–4.

[6] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment," *Sensors*, vol. 19, no. 4, p. 838, Feb 2019. [Online]. Available: http://dx.doi.org/10.3390/s19040838

[7] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.", 2016.

[8] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 28–31.

[9] S. Lim, S. Kim, and D. Kim, "Performance effect analysis for insect classification using convolutional neural network," in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, 2017, pp. 210–215.

[10] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, "Insect detection and classification based on an improved convolutional neural network," *Sensors*, vol. 18, no. 12, p. 4169, 2018.

[11] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, 2014.

[12] T. Polonelli, D. Battistini, M. Rusci, D. Brunelli, and L. Benini, "An energy optimized jpeg encoder for parallel ultra-low-power," *Applications in Electronics Pervading Industry, Environment and Society: APPLEPIES 2019*, vol. 627, p. 125, 2020.

[13] T. Polonelli, D. Brunelli, and L. Benini, "Slotted ALOHA overlay on LoRaWAN - a distributed synchronization approach," in *2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2018, pp. 129–132.

[14] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on lora and lorawan," *Sensors*, vol. 18, no. 7, p. 2104, 2018.