Energy Neutral Machine Learning based IoT device for Pest detection in Precision Agriculture

Davide Brunelli, Andrea Albanese, Donato d'Acunto and Matteo Nardello
Department of Industrial Engineering
University of Trento, Italy
Trento, Italy
email:{name.surname}@unitn.it

Abstract

Apples are among the topmost fruit crops of the world, and apple orchards are widely expanding in many regions and countries. The most common problem for these crops is the attack of the Codling Moth, which is a dangerous parasite for apples. IoT sensing devices can nowadays run near sensor machine learning algorithms, thus giving not only the possibility of collecting data over wide coverage but featuring even immediate data analysis and anomaly detection. Near sensor neural network algorithms can automatically detect the Codling Moth: the system takes a picture of the trap, preprocesses it, and crops each insect for classification and eventually sends a notification to the farmer if any Codling Moth is detected. The application is developed on a low energy platform powered by a solar panel of a few hundred cm^2, realizing an energy autonomous system capable of operating unattended continually over low power wide area networks (LPWANs). An insightful aspect of this IoT solution is the low power platform for a machine learning algorithm used for IoT fast prototyping. The hardware is based on the Raspberry Pi3 board and the Intel Movidius Neural Compute Stick, responsible for the preprocessing technique and the neural network implementation, respectively. The network model has been analyzed in detail, showing parameter settings and the limitations for the specific hardware constraints. The performance of the proposed system is assessed, and remarks about power consumption are discussed for achieving the zero energy balance of the system.

Index Terms

Internet of Things, Machine Learning, Computer Vision, Precision Agriculture, Codling Moth, Neural Network, Trap images, Energy Harvesting

I. Introduction

Recent technological advances have paved the way for remote agricultural sensing and automation. Consequently, sophisticated energy neutral low cost sensors [1] and communication systems [2] can be used as components to monitor and control systems for a sustainable and healthy environment, which is a requirement for smart agriculture applications [3]. However, current wireless sensing platforms and communication systems are designed for bare remote monitoring without taking any immediate decision after the damage has already been done [4]. Moreover, the large scale deployment of sensors would result in a tremendous increase in the number of connections and the amount of data to be transmitted, which could overwhelm current communication systems and also data analysis algorithms [5].

Reconceiving the paradigm of remote sensing operation is imperative to improve the operational performance of precision agriculture. Adding intelligence to the nodes, shifting the detection of anomalies near the sensor to permit decisions and actions as soon as possible, is the key to reduce the communication costs, latencies and to permit high scalability of IoT solutions in agricultural environments.

Nowadays, Machine Learning (ML) algorithms are widely used in many fields and are particularly innovative in agriculture to compute tasks such as species recognition [6], water management, crop quality [7], disease detection, and weed detection.

This article focuses on an automatic method for monitoring parasite insects from images taken in pest traps. The *codling moth* is a particular insect that looks like a butterfly, and it is a dangerous parasite for apple fruit crops. An energy efficient IoT solution shows how the feasibility of classifying parasites from other general insects autonomously, using low power consumption hardware directly infield. Moreover, the article shows the fast and cost effective realization of an intelligent sensor and communication system that can be applied in agricultural monitoring and control. It runs machine learning on the sensor board and, if the insect captured by the camera is classified as a *codling moth*, a report is sent for an immediate counteraction.

II. IoT system

Current methods to monitor the pests consist of capturing insects using commercial pheromone based glue traps, as shown in Fig. 1a, that attract insects even if present at very low densities. Periodic in field inspections or simple wireless cameras permit the farmer to watch each insect and determine if it is a *codling moth* or not [8]. This process is not as smart as an IoT solution could be. In fact, it is slow because it requires the full time presence of an expert, and it is inefficient because, even though machine learning is used, it requires full images sent for remote classification [6].

(a) Commercial trap.

(b) Prototype of the IoT neural network codling moth smart trap.

Fig. 1: Codling moth traps

The proposed system, as shown in Fig. 1b, processes the picture in situ near the sensor (preprocessing algorithm), returns a classification of the insects (machine learning algorithm) in the trap, and eventually it sends a notification to the farmer if it recognizes a *codling moth*.

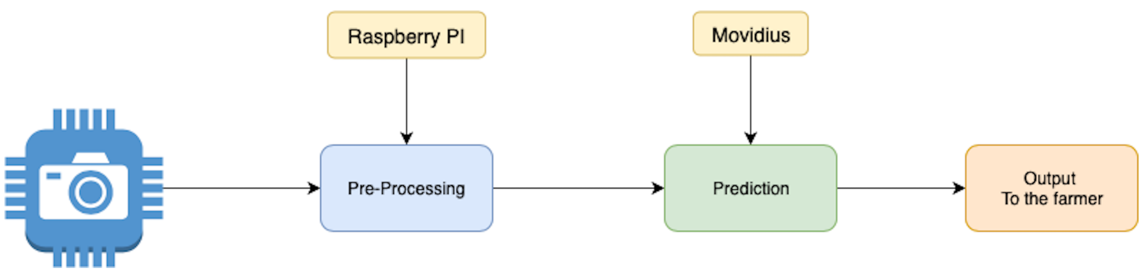


Fig. 2: Overall system diagram.

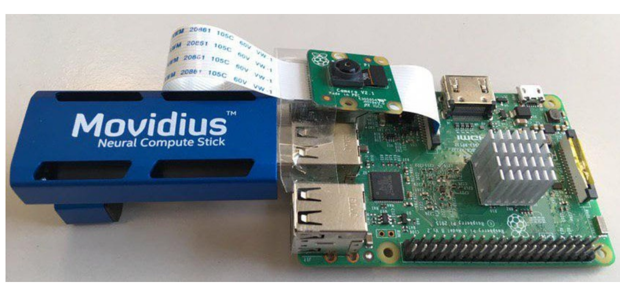


Fig. 3: Hardware implementation.

As presented in Fig. 2, the system is embedded on a Raspberry PI 3 that provides the preprocessing stage. Then an Intel Movidius Neural Compute Stick (NCS) with an Intel Myriad X neural accelerator as a Vision Processing Unit (VPU) classifies the images using the model obtained after the training of the Deep Neural Network (DNN).

The system, shown in Fig. 3, has been designed to bring IoT technologies in agriculture where the need to collect the output over vast areas requires long range communication. Thanks to the onboard intelligence, the output of the smart trap is limited to the few bytes for the report after the classification process, and output messages can be managed even with low bitrates. In case the farmer needs a visual confirmation from the captured picture, few images per day can be transmitted as well. Therefore the trap uses Low Power Wide Area Network (LPWAN) technologies and specifically the LoRaWAN protocol that has gained momentum in the market recently. LoRa is a wireless modulation designed for long range communication at very low energy consumption and bitrate [9]. LoRaWAN stack defines the communication and security protocols to guarantee interoperability on top of the LoRa network [10], [11].

Image preprocessing and deep learning

The data set used to start the DNN training contained approximately 1300 pictures and has been incremented when more insects have been trapped during the earliest experiments. The dataset represents two classes: *codling moth* and *general insects*. These figures are used to feed and train the DNN with a TensorFlow model. We used the VGG16 model, developed by Oxford University [12]. Then it has been converted to a graph model used to perform the classification on the VPU.

The dataset was created with the same camera and trap. The camera captures the bottom side of the insect glue trap, thus as shown in Figure 4, pictures may contain a high number of trapped insects to classify. Thus, the images are processed in situ to separate each insect in sub tiles from the original picture. This step is essential since it filters the raw pictures, as shown in Figure 4, and produces tiles that contain only one insect. This algorithm is used in two different cases:

- To build a large and comprehensive dataset of pictures for training the DNN model. We started with a hundred of raw pictures, that generated more than 1300 tiles containing only one insect.
- At each application startup, a picture of the trap is taken firstly, and then, thanks to the preprocessing algorithm, each new trapped insect is cropped for the classification step.

The task efficiently exploits features such as color (dark subjects on white background) and the shape of the insects with a Blob Extraction algorithm. The process for image crop consists in:

- Conversion of the frame from RGB to GRAY scale;
- Smoothing (or blurring) of the frame with a Gaussian filter;
- Edge extraction through Canny operator;
- Some dilation and erosion of the picture.

After these operators, the blobs are detected through the OpenCV blob detector. Then each blob is collected in a vector and the corresponding regions of interest are cropped. All the new pictures are saved for the neural network classification.

III. Training, Validation and Test

For the training stage, we used the rapid development of neural networks for image classification provided by the TensorFlow library [13]. In a machine learning approach, an initial training step is required. The training consists of an offline process that optimizes the neural network using a large dataset of labeled images. In this way, the system learns the classes assigned to the images. The basic unit of a DNN is the neuron (or node) that multiplies by weight values the input signals. The training phase adjusts the weight values, while some parameters, such as the number of epochs and the image size, can improve the accuracy of a DNN. Epochs represent the number of times all of the training vectors are used once to update the weights. Each epoch finishes with a validation step that evaluates the ongoing training process. A good tradeoff between the number of epochs and image size is necessary for a correct training stage and to meet the hardware constraints. The training stage of this application has been assessed with three different configurations:

- 75 epochs, image size 224×224;
- 10 epochs, image size 112×112;
- 10 epochs, image size 52×52;

The results of the training tests are presented in Fig. 5.

Notice that training and validation accuracy using 75 epochs is going to be saturated that suggests that the number of epochs can be decreased to achieve similar performance. As shown in the graphs, ten epochs are enough for the target accuracy. Moreover, to avoid possible overflow in the Movidius NCS and to save memory on the Raspberry

(a) Raw picture.

(b) Tile with a Codling Moth. (c) Tile with a general insect. (d) Tile with a Codling Moth.

Fig. 4: Examples of cropped images after pre-processing

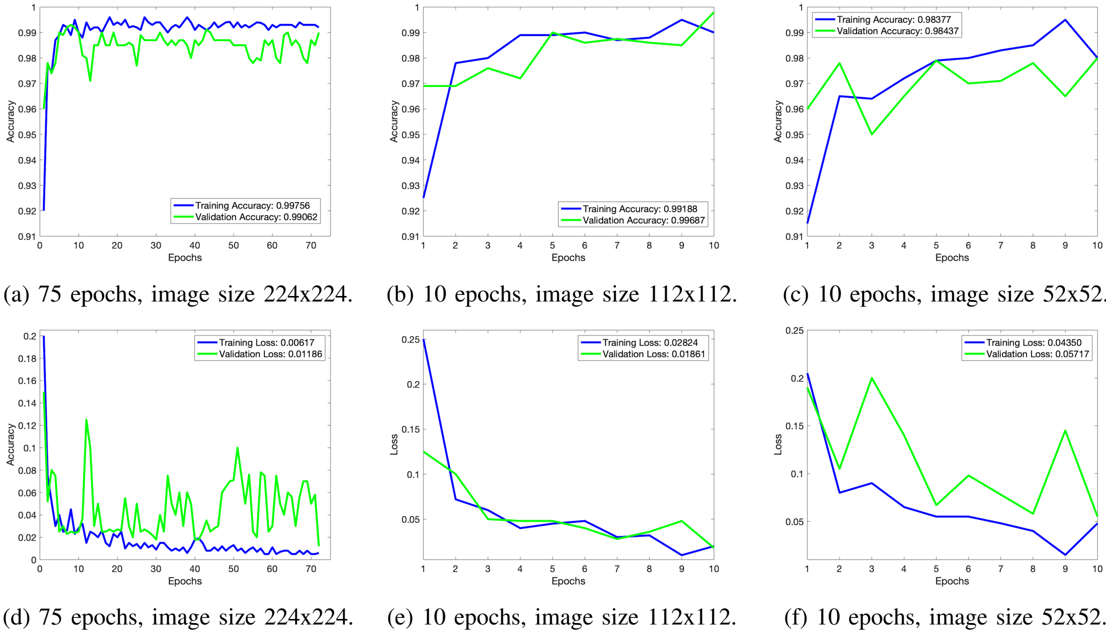
Pi 3, the image size can be decreased to meet the hardware constraints because we can use simpler models. We used and tested images of 112×112 and 52×52 pixels size, as shown in Fig. 5. Small size images, clearly, shows worse performance with respect to bigger tiles. Nevertheless, about 98% accuracy has been achieved, that satisfies the requirements expected by farmers for an IoT service of parasites monitoring.

Fig. 6 shows an example of the output from the classification. The DNN provides a confidence measure that indicates how close the detected object is to a general insect or the target *Codling Moth*.

The tests of the DNN model were carried out during 12 weeks in an apple orchard with the insect glue trap shown in Fig. 1. Tests have involved 262 news insects where:

- 80.6% was classified correctly;
- 4.8% was false positives;
- 6.4% was false negatives;
- 8.2% was uncertain;

Thus the precision is $94,38\%$, the recall is $92,6\%$, and only $8,2\%$ need a user assessment watching the raw image.



(a) 75 epochs, image size 224x224. (b) 10 epochs, image size 112x112. (c) 10 epochs, image size 52x52.

(d) 75 epochs, image size 224x224. (e) 10 epochs, image size 112x112. (f) 10 epochs, image size 52x52.

Fig. 5: Training and validation accuracy and loss function.

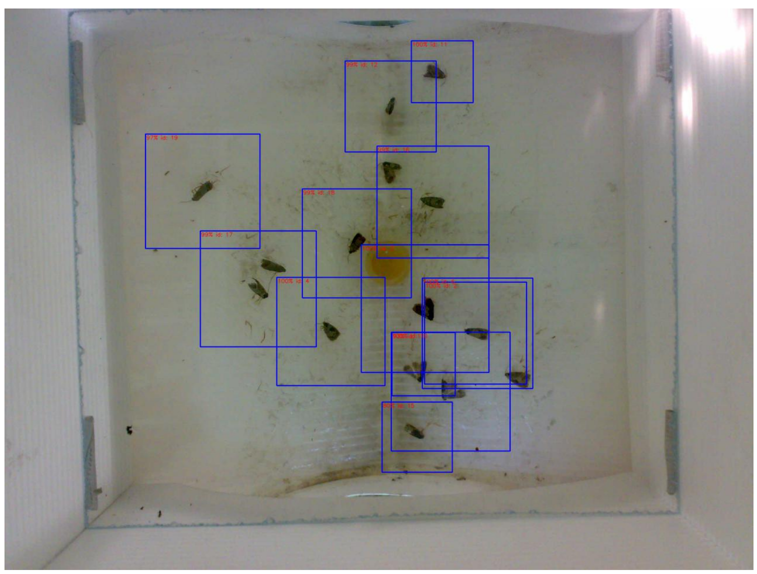


Fig. 6: Example of moth detection from the system.

IV. Power assessment

In apple orchards, codling moth checking is usually executed twice every day. We evaluated the power consumption of the overall system's classification, as shown in Fig. 3, which is divided into five general tasks with different execution time and current consumption:

- Task 0: Boot of the Raspberry (Time 43.68 s, Average Current 345 mA);
- Task 1: Image capture (Time 3.45 s, Average Current 394 mA);
- Task 2: Preprocessing (Time 4.07 s, Average Current 501 mA);
- Task 3: Classification (Time 10.19 s, Average Current 525 mA);
- Task 4: Report/Alarm generation (Time 0.34 s, Average Current 525 mA).

When the system finishes the Task 4, it shuts down, and zeroing its power consumption; while a nW real time clock (RTC) is activated to trigger and boot the application when planned.

As expected, it is possible to observe that T3 is the most power hungry task because it combines the usage of the Raspberry and the Intel Movidius. Fig. 7 shows the power consumption of the overall system from T0 to T4, and the total energy necessary is $124.1J$ and thus a $9000mAh$ battery is sufficient to sustain the system for more than one year. Moreover, combining the system with a $0.5W$ solar panel of a few hundred cm^2, as presented in [14], the energy intake will be enough to permit to the smart camera to operate unattended indefinitely.

This particular aspect represents a breakthrough for agricultural activities because this means that a farmer could use a smart IoT insect trap, forgetting about its maintenance, and waiting for only automatic alerts if some *Codling Moth* is captured.

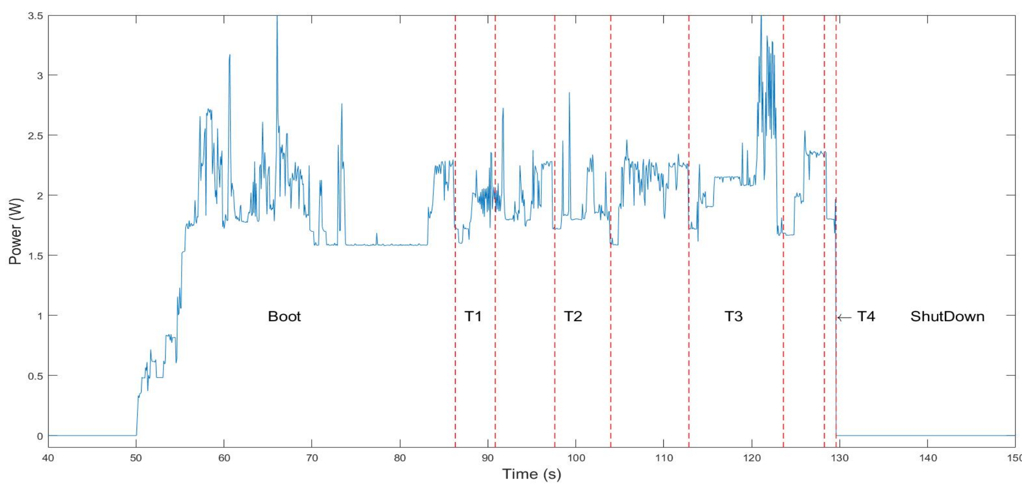


Fig. 7: System power consumption.

V. Conclusions

Even though the proposed system does not use ultra low power microprocessors or microcontrollers, its average power consumption is minimal, because of its low duty cycle.

Due to the low cost of the hardware, this type of system can scale to several installations in the farmer's apple orchard, and save time and money for human intervention in trap checking every day. This type of application is straightforward, innovative, and it gives an additional value to agriculture. In this way, it is possible to use treatments for *Codling Moth* only when the system detects threats for crops, optimizing the use of chemicals, and mitigating their impact on the environment.

Acknowledgement

This research was supported by the IoT Rapid-Proto Labs project, funded by Erasmus+ Knowledge Alliances program of the European Union (588386-EPP-1-2017-FI-EPPKA2-KA).

References

[1] D. Balsamo, A. Das, A. S. Weddell, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini, "Graceful performance modulation for power-neutral transient computing systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 5, pp. 738–749, May 2016.

[2] 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)*, Oct 2018, pp. 129–132.

[3] M. Rossi and D. Brunelli, "Ultra low power wireless gas sensor network for environmental monitoring applications," in *2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*, Sep. 2012, pp. 75–81.

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

[5] D. Brunelli and C. Caione, "Sparse recovery optimization in wireless sensor networks with a sub-nyquist sampling rate," *Sensors*, vol. 15, no. 7, pp. 16 654–16 673, 2015.

[6] 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.

[7] M. Guermandi, S. Benatti, D. Brunelli, V. Kartsch, and L. Benini, "Towards a wearable interface for food quality grading through erp analysis," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1–5.
[8] A. Guarnieri, G. Molari, and V. Rondelli, "Sistema elettronico per il monitoraggio di insetti dannosi in frutteto," in *Gestione e controllo dei sistemi agrari e forestali – Memorie AIIA 2011*, 2011.
[9] L. Tessaro, C. Raffaldi, M. Rossi, and D. Brunelli, "Lightweight synchronization algorithm with self-calibration for industrial lora sensor networks," in *2018 Workshop on Metrology for Industry 4.0 and IoT*, April 2018, pp. 259–263.
[10] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, Sep. 2017.
[11] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted aloha on lorawan-design, analysis, and deployment," *Sensors*, vol. 19, no. 4, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/4/838
[12] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2015, pp. 730–734.
[13] NeuralNetworks, https://github.com/frank1789/NeuralNetworks, online; accessed 25 May 2019.
[14] D. Brunelli, D. Dondi, A. Bertacchini, L. Larcher, P. Pavan, and L. Benini, "Photovoltaic scavenging systems: Modeling and optimization," *Microelectronics Journal*, vol. 40, no. 9, pp. 1337–1344, 2009.