Neural networks for Pest Detection in Precision Agriculture

Andrea Segalla, Gianluca Fiacco, Luca Tramarin, Matteo Nardello and Davide Brunelli
Department of Industrial Engineering, University of Trento, Italy
{name.surname}@unitn.it

***Abstract*—Apple is one of the most produced fruit crops in the world. Recent advances in Artificial Intelligence and the Internet of Things can reduce production costs and improve crop quality by providing prompt detection of dangerous parasites. This paper presents an effective solution to automate the detection of the Codling Moths. The system takes pictures of trapped insects in the orchard, analyzes them through a DNN algorithm, and sends alarms to the farmer in case of a positive detection. The system is fully autonomous and can operate unattended for the entire crop season. Detection reports are used for optimizing the treatment with chemicals only when threats are identified. The prototype is designed with an embedded platform powered by a small solar panel to achieve an energy-neutral balance.**

I. Introduction

With recent advances in Artificial Intelligence and the Internet of Things (IoT), the number of energy autonomous sensors and devices that can be implemented in agricultural solutions has grown enormously [1]. This growth and accessibility have favored the emergence of IoT and Cloud solutions, giving rise to a phenomenon known as Smart Farming [2].

Deep Neural Networks (DNNs) have become predominant for vision recognition and other patterns detection in recent years, even in ultra-low power cameras [3]. They are more recently used to build robust analytic tools for Smart Farming [4]–[6] because crop losses and pest control can be extremely expensive.

Deep learning is a promising approach for extracting accurate information from raw sensor data from IoT devices deployed in complex environments. In conventional cloud computing, all data must be uploaded to centralized servers, and, after computation, the results need to be sent back to sensors, devices, or actuator. This process creates high pressure on the network, specifically in the data transmission costs of bandwidth and energy resources [7].

Deep learning can enable IoT devices to interpret unstructured multimedia data, and intelligently react to user and environmental events. Adding intelligence to the nodes and shifting the decision of anomalies near the sensor allows fast decisions and actions and permit high scalability. Moreover, local data processing is the key to reduce communication costs and latencies as an alternative to data compression [8].

Monitoring the number of insect pests is crucial in pheromone-based pest management systems as pest infestation is one of the main factors that affect harvest losses [9]–[11]. The most common method of measuring insect infestations is to identify and count the insects manually; captured digital images are analyzed by human experts, or farmers, to recognize and count pests. However, visual inspection is labor-intensive and inefficient; therefore, subjective factors can affect the accuracy of population counts. With the development of information technology, the researchers have proposed to use computer vision techniques for automatically identifying and counting agricultural pests. The same digital images used for manually identifying pest infestation are being used to train machine learning algorithms for automatic disease detection [5], [12]–[16]. Once trained, intelligent visual IoT devices can be deployed directly in orchards for autonomously monitoring dangerous parasites.

This paper presents the study of an AI-IoT smart device for pest detection in Smart Farming. The device is based on a low-power platform and integrating energy harvesting capabilities, and can be left operating inside common pheromone-based traps for the entire crop season. A report is remotely sent thanks to a long-range low-power LoRa radio in case of a pest infestation.

The paper is organized as follows: Section II briefly presents the hardware overview of the system and the software state machine, while the machine learning task is discussed in section III. The overall evaluation of the system is presented and discussed in section IV. Here we characterize the networks' accuracy and the energy requirements of the different IoT device tasks. Closing this work, Section V discusses future improvements and draws conclusions.

II. System Overview

Figure 1 presents the system architecture. The prototype is based on a Raspberry-Pi platform and integrates a camera for acquiring trap images and an Intel Neural Compute Stick for evaluating the DNN. Long-range connectivity is guaranteed by a LoRa modem perfectly suitable for low bitrate communications. To extend the battery life of the system, the prototype also integrates a complete solar energy harvester, as other solar-harvesting wireless smart cameras [17]. In this early stage it is implemented using a Pi-Juice-Hat [18] that provides all the components for managing a low-power platform. More specifically, a low-power RTC clock is used to wake up the system and a voltage monitor to ensure enough energy is stored in the battery. The prototype developed is compact (101×67×55 mm), portable, and easy to install. The device can be installed inside common pheromone-based traps. Codling moth detection is executed twice a day. A picture of the trapped insects is collected and evaluated. Figure 2 presents the trap where the prototype was tested and evaluated.

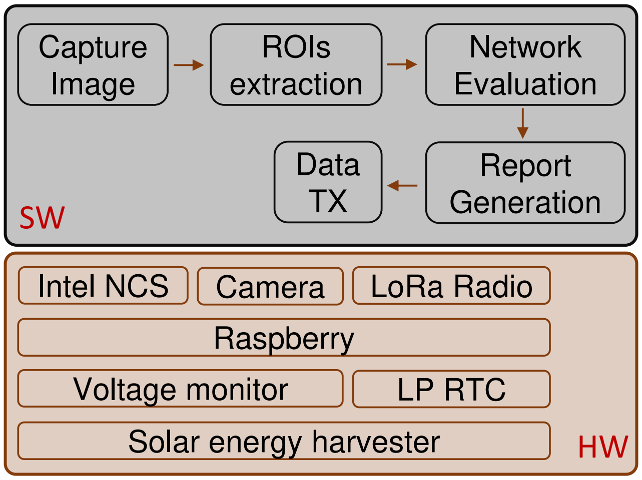


Fig. 1: System architecture

The software state machine is simple and encompasses just a few steps. The traps' images are acquired using the Camera, adapted using the OpenCV library, and used as an input to the neural network for recognition. Network evaluation speed is improved using the Intel Movidius Neural Compute Stick (NCS) [19], a tiny fanless deep-learning device used to evaluate DNNs at the edge. After the DNN assessment, a report is generated and sent using the LoRa modem to the farmer. Two different neural networks are evaluated to compare this kind of application's performance.

Fig. 2: Smart Trap installed in an apple orchard

III. Neural Network

DNNs, or deep learning, refer to a specific class of neural networks algorithms. Learning tasks are broken down and distributed onto machine learning algorithms organized in consecutive layers. Together the layers constitute an artificial neural network that mimics the distributed approach to problem-solving carried out by neurons in a human brain.

Two CNNs were chosen for this application and then evaluated their performance. The first is **LeNet** [20], a multilayer neural network trained with back-propagation. It is designed to recognize visual patterns directly from pixel images with minimal preprocessing. By modern standards, LeNet is straightforward. It is made up of 7 layers. The layer composition consists of 3 convolutional (C1, C3 and C5) layers, 2 subsampling layers (S2 and S4) and 1 fully connected layer (F6), that are followed by the output layer. To reduce the computing demanding of the network and to permit convolutional kernels to learn different patterns, individual convolutional kernels, in the layer C3, do not use all of the features generated by the layer S2. This allows the network to learn the best internal representation from raw images automatically. These key features perfectly fit the requirements for evaluating insect pictures because the network can automatically learn different insect patterns.

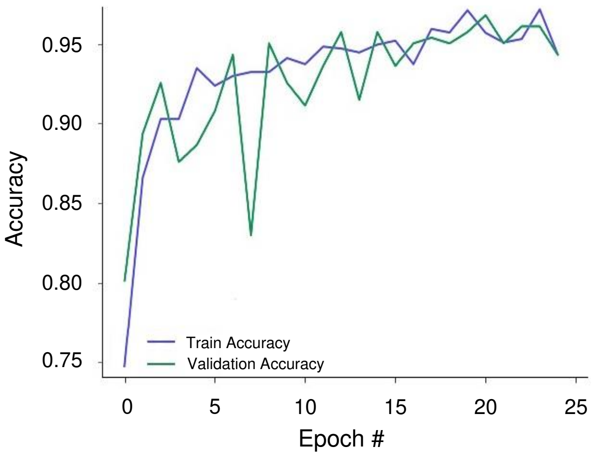
The second trained network is based on **VGG16** [21], a very deep Convolutional Neural Network developed mainly for face recognition. Instead of having many hyper-parameters, VGG16 focused on having convolution layers of 3x3 filter with a stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2. The network is arranged in convolution and max pool layers consistently throughout the whole architecture. It concludes with 2 fully connected layers followed by a softmax for output. Convolutional layers do "feature extraction," acting as perception fields, finding patterns and geometrical shapes of progressing complexity, while fully connected layers act as a classical perceptron, classifying objects based on what shapes were present in the image. This network is pretty large, and it has millions of parameters. However, it was selected both to understand how accuracy can vary with more complex models, due to the intrinsic characteristic of self-tuning for different complex shapes, like the case of insects.

A. Related work

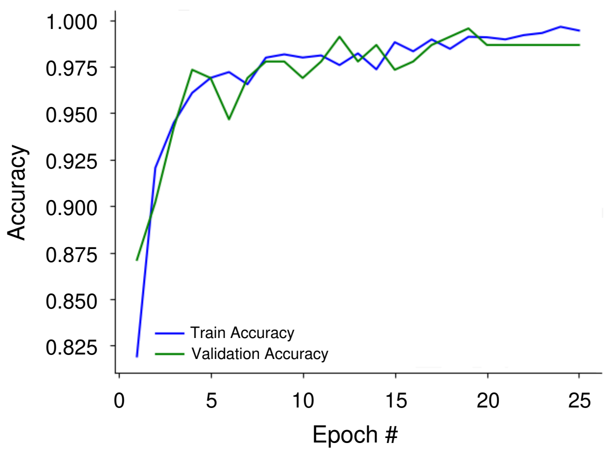
Effective crop protection requires early and accurate detection of biotic stress. The identification and monitoring of insect pests using automatic traps bring a novel approach to the integrated pest management [11], [22].

Plenty of previous work has considered autonomous insect detection and classification using machine learning. In a recent approach [23], deep learning object detectors were applied to create detection models for three species of moths collected from pheromone traps. The best model showed an average accuracy of over 90% in detecting 4 different insect classes. Even if the approach provides good accuracy, it cannot be deployed inside a low-power embedded platform due to the network's complexity.

Another research [24] investigated the potential of using hyperspectral imaging in the spectral region between the VS and NIR regions (400–1000 nm) for assessing codling moth



(a)



(b)

Fig. 3: Training and validation accuracy for LeNet (3a) and VGG16 (3b)

infestation inside apples. Several classification and data types were conducted, and the best-achieved classification rate is about 82%. Furthermore, this approach targets the detection after the apples have been harvested, and cannot be deployed in orchards.

An approach similar to the one discussed in this paper is investigated in [25] and [26]. There, convolutional neural networks are used to classify and detect insects. Both implementations show how a CNN can be a good choice for automating pest detection with an average accuracy of near 90%, also in the case of multiple insect classes.

Even if the approach presented in this paper is not a novelty, the proposed solution can provide state-of-the-art detection results directly from apple orchards. Our system provides high accuracy for the implemented task, without the need to first upload acquired data to the cloud. All the computation is carried out on the edge, with the capability to exploit energy harvesting.

B. Training

To train the two proposed models for codling moth detection, two different approaches were used. The LeNet model was trained using a mobile workstation, while for the VGG16 model an HPC was used. This is due to the fact that the VGG16 model is more complex and requires higher computing power to complete the training phase. For reference, in both cases, the training phase was completed in less than 10 minutes.

The data set used during the training consists of 1200 images, divided into two classes. 800 containing the codling moth (C1) and 400 not containing the codling moth (C2). The data set was further divided into 900 images used for the training (645 C1 - 255 C2, images) and 300 for the validation (200 C1 - 100 C2, images)

Regarding image dimension, LeNet model is trained using 28x28 pixels tiles while VGG16 uses 56x56 tiles.

Training and validation accuracy are presented in Figure 3. Analyzing the results, we can see in the first one an irregular increment of precision during the validation. This sudden loss of precision is due to the less robustness of LeNet with respect to VGG16. However, at the end of the validation process, both networks achieve an acceptable level of precision: LeNet about 94% and VGG16 about 99%. Future works should extend the data set with more pictures, encompassing different insects than the Codling moth. Figure 4 presents an example of a ROI extracted by the learning algorithm.

Fig. 4: Example of a ROI extracted by the learning algorithm from a picture captured inside a pheromone-based common trap

IV. Evaluation and Discussion

Table I present the performance of the two CNN implemented. Using these results, it is possible to compute the

TABLE I: CNNs prediction results using a batch of 119 images.

	Ture Positive	**True Negative**	**False Positive**	**False Negative**
LeNet	83	34	2	0
VGG16	71	34	2	12

performance of the neural networks in terms of accuracy, recall, precision and f-measure.

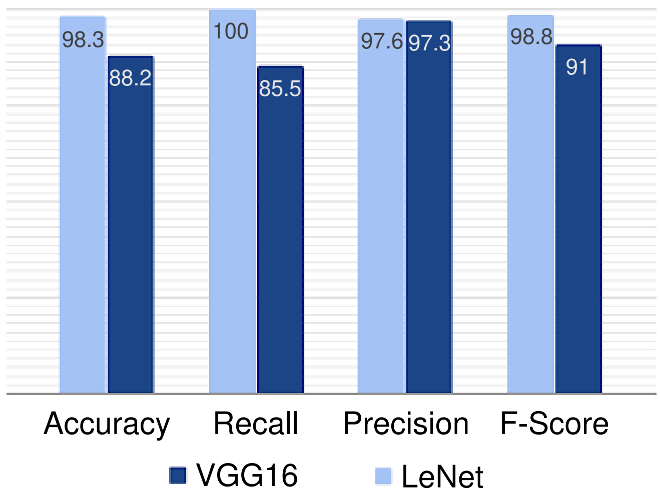


Fig. 5: Figures of merit for both LeNet and VGG16 CNNs

The histograms presented in Figure 5 show the results obtained and allow us to compare the two CNNs rapidly. In VGG16, we can see a high precision and a low recall, meaning that it misses some moths while the predicted ones are correct. Consequently, the F-Measure is lower. LeNet presents good results in all the four parameters shown above. For this reason, it is possible to conclude that LeNet outperforms VGG16 in this application scenario.

VGG16 lower performance is probably associated with the reduced dataset used during the training phase. Due to the network architecture and the number of parameters to adjust, the training phase needs more data to produce reliable results. Future works should use a larger dataset with many different insects to obtain more reliable and more truthful results for a real application.

A. *Tasks characterization*

The system is kept in sleep mode most of the time to lower the energy requirement of the IoT node, The low-power RTC, integrated into the PiJuiceHAT, wakes the system up only when needed to execute the tasks whenever planned.

To characterize the system's energy budget, we divided the software pipeline into four different tasks and then evaluated both time duration and energy demand. The four tasks are: *1) Image acquisition; 2) pre-processing; 3) Inference; 4) Data Transmission*. During these tests, all the unnecessary functionalities have been disabled (i.e., like Wi-Fi, Bluetooth, graphic interface of Raspbian OS).

Three different prototypes have been tested

- Raspberry Pi3 evaluating a LeNet model;
- Raspberry Pi3 evaluating a VGG16 model using an Intel NCS;
- Raspberry Pi4 evaluating a LeNet model.

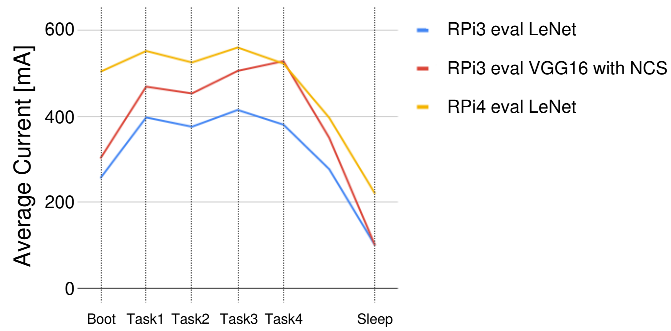


Fig. 6: Average current consumption of each task. The models using Raspberry Pi4 consume more current in each task with respect to the other models, except for the third and fourth tasks (Neural network inference and message transmission) of the Raspberry Pi3 with VGG16 and Movidius.

Results are presented in table II. As expected, experiments using the Movidius are accelerated than the ones without it. Indeed, the most significant difference is observable in task 3, which requires the highest time because it involves the neural network evaluation. This task is the most critical also from the power consumption. In particular, the models with VGG16 and Movidius are more expensive in terms of energy requirements. Task 1 and Task 4 consist of shorter computation and need similar time and consumption in all the different prototypes.

After the complete evaluation of the Pi3 running both models, LeNet has been loaded on the Raspberry Pi4 to study the difference, in terms of performance, between Pi3 and Pi4. It has been observed that the configuration of Raspberry Pi4 with LeNet consumes more than the Raspberry Pi3 with LeNet. For this reason, the setup with Raspberry Pi4 with VGG16 on Movidius was not analyzed. As can be noted in Figure 6 results using a Pi4 are in general worse than models developed on Pi3.

Concerning energy consumption, the best configuration is the Raspberry PI3 optimized with VGG16 + Movidius. It is due to the use of the Movidius that improves the performance of the neural network inference reducing the task execution times and the total energy, even if the average power is higher. The current trace collected while evaluating this implementation is presented in Figure 7. Indeed, thanks to Movidius accelerator, the VGG16 uploading model is faster than the LeNet, even if the latter occupies much less more resources (1 GB for VGG16 vs. 15 MB for LeNet). Model uploading is the most time-consuming part of task 3. In terms of time and energy, a possible optimization can be running the LeNet evaluation using the NCS. Unfortunately, the model is not directly compatible with the NCS and needs further processing.

B. *Energy Harvesting and battery life*

To assess the expected battery life, the first test calculated the number of complete cycles (i.e., from capturing an image to sending the report using the LoRa modem) achievable with

TABLE II: Average current consumption and time duration for the four tasks presented in subsection IV-A. Three different implementations have been tested: **1)** Raspberry Pi3 evaluating LeNet; **2)** Raspberry Pi3 + Movidius evaluating VGG16; **3)** Raspberry Pi4 evaluating LeNet

	Task 1		**Task 2**		**Task 3**		**Task4**	
	Duration [s]	**Consumption [mA]**	**Duration [s]**	**Consumption [mA]**	**Duration [s]**	**Consumption [mA]**	**Duration [s]**	**Consumption [mA]**
1	1.41	397.1	3.51	375.6	33.05	414.5	1.13	380.4
2	0.48	468.7	3.4	452.9	16.30	505.5	0.86	527.9
3	1.52	551.8	2.57	524.9	18.82	559.8	0.70	522.3

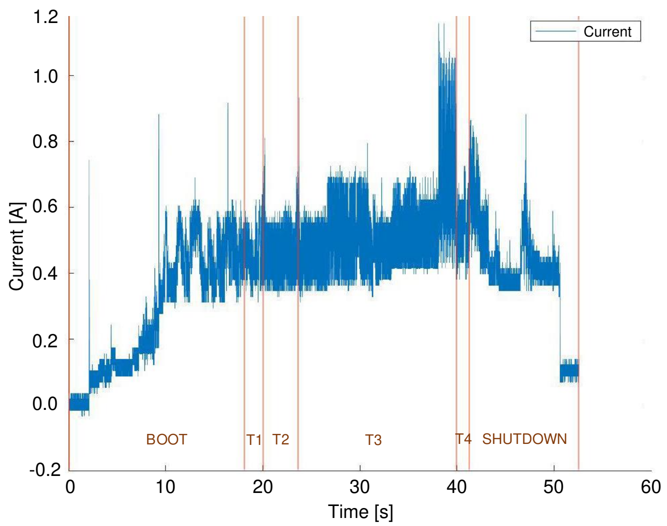


Fig. 7: Current consumption trace of a complete cycle using a RPI3 with Intel NCS evaluating a VGG16 network.

the integrated battery, a single cell LiPo battery providing 1820 mAh.

Taking into consideration the setup that exhibits the maximum energy demand – Raspberry Pi4 evaluating a LeNet model – we can achieve 120 cycles. On the contrary, considering the best implementation – Raspberry Pi3 evaluating the VGG16 model using the NCS – we can achieve a maximum of 169 cycles. If we assume that a farmer usually checks the traps for pest infestation twice a day, the implemented system expected battery life goes from a minimum of 60 days to a maximum of 84. Table III presents the result of this analysis.

However, the proposed smart device also integrates a small (140 x 100 mm) solar panel providing a maximum of 0.8 Wp. Using this solar panel, we can recharge the battery from zero in around 3 hours (when working in optimal condition [27]).

TABLE III: Average energy consumption and expected battery life for: **1)** Raspberry Pi3 evaluating LeNet; **2)** Raspberry Pi3 + Movidius evaluating VGG16; **3)** Raspberry Pi4 evaluating LeNet

	RPi average current consumption for 1 cycle [mAh]	**Battery average current consumption for 1 cycle [mAh]**	**Battery average energy for 1 cycle [J]**	**Number of cycles using the integrated battery**
1	6.94	12.55	166.6	146
2	5.96	10.73	143	169
3	8.35	15.04	200.4	120

Considering the amount of energy required for completing 2 cycles (i.e., one day of operations), we can safely say that the platform is energy neutral and can self-sustain its operations. Other energy harvesting solutions could be used, for example the one that exploit microbial fuel cells [28], or introducing Intermittent computing [29], [30].

C. ROIs images transmission

Eventually, tests are conducted to measure the energy required when the extracted Region of Interest (ROI) is transmitted wirelessly. We considered 10 frames, 60×60, ROIs, and compressed using JPEG. The information consisting of 26,560 bytes was transmitted using the LoRa modem, using 111 sequential packets. Considering a delay of 0.1 s between two consecutive packets, the time of task 4 was 115.32s. Updating this value, the average consumption of this configuration becomes 30.47 mAh requiring 406 J. In this case, the node can still complete 59 cycles, meaning about a month of expected Pi-Juice-Hat battery life.

V. Conclusion

Deep learning is nowadays a state-of-the-art approach for extracting accurate information from raw sensor data in many applications. This paper presented the study of an effective solution for pest detection in precision agriculture. A range of implementations and two different DNN models have been studied. Results show how it is possible to run learning tasks directly on the sensor node, allowing the automation of codling moth detection task. Only the analysis results are transmitted, reducing the bandwidth and energy costs associated with wireless transmission. Moreover, we demonstrated energy neutrality by combining a small solar panel, making the device fully autonomous.

Acknowledgement

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

References

[1] D. Brunelli, P. Tosato, and M. Rossi, “Flora health wireless monitoring with plant-microbial fuel cell,” *Procedia Engineering*, vol. 168, pp. 1646–1650, 2016.

[2] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, “Big data in smart farming – a review,” *Agricultural Systems*, vol. 153, pp. 69 – 80, 2017.

[3] M. Nardello, H. Desai, D. Brunelli, and B. Lucia, "Camaroptera: A batteryless long-range remote visual sensing system," in *Proceedings of the 7th International Workshop on Energy Harvesting Energy-Neutral Sensing Systems*, ser. ENSsys'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 8–14.

[4] R. S. Alonso, I. Sittón-Candanedo, Óscar García, J. Prieto, and S. Rodríguez-González, "An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario," *Ad Hoc Networks*, vol. 98, p. 102047, 2020.

[5] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Computers and Electronics in Agriculture*, vol. 123, pp. 17 – 28, 2016.

[6] O. Debauche, S. Mahmoudi, M. Elmoulat, S. A. Mahmoudi, P. Manneback, and F. Lebeau, "Edge ai-iot pivot irrigation, plant diseases and pests identification," *Procedia Computer Science*, 2020.

[7] P. Patel, M. Intizar Ali, and A. Sheth, "On using the intelligent edge for iot analytics," *IEEE Intelligent Systems*, vol. 32, no. 5, pp. 64–69, Sep. 2017.

[8] D. Brunelli and C. Caione, "Sparse recovery optimization in wireless sensor networks with a sub-nyquist sampling rate," *Sensors*, vol. 15, no. 7, p. 16654–16673, Jul 2015.

[9] S. S. Patil and S. A. Thorat, "Early detection of grapes diseases using machine learning and iot," in *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, Aug 2016, pp. 1–5.

[10] A. Albanese, D. d'Acunto, and D. Brunelli, "Pest detection for precision agriculture based on iot machine learning," *Lecture Notes in Electrical Engineering*, vol. 627, pp. 65–72, 2020.

[11] J. Behmann, A.-K. Mahlein, T. Rumpf, C. Römer, and L. Plümer, "A review of advanced machine learning methods for the detection of biotic stress in precision crop protection," *Precision Agriculture*, vol. 16, no. 3, pp. 239–260, 2015. [Online]. Available: https://doi.org/10.1007/s11119-014-9372-7

[12] 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, December 2019.

[13] T. Liu, W. Chen, W. Wu, C. Sun, W. Guo, and X. Zhu, "Detection of aphids in wheat fields using a computer vision technique," *Biosystems Engineering*, vol. 141, pp. 82 – 93, 2016.

[14] Q. YAO, J. LV, Q. jie LIU, G. qiang DIAO, B. jun YANG, H. ming CHEN, and J. TANG, "An insect imaging system to automate rice light-trap pest identification," *Journal of Integrative Agriculture*, vol. 11, no. 6, pp. 978 – 985, 2012.

[15] 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.

[16] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen, and H. Chen, "Multi-level learning features for automatic classification of field crop pests," *Computers and Electronics in Agriculture*, vol. 152, pp. 233 – 241, 2018.

[17] M. Magno, D. Brunelli, L. Thiele, and L. Benini, "Adaptive power control for solar harvesting multimodal wireless smart camera," in *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2009, pp. 1–7.

[18] PiSupply. https://github.com/pisupply/pijuice. [Online]. Available: https://github.com/PiSupply/PiJuice

[19] Intel. (2019) Intel movidius neural compute stick. [Online]. Available: https://movidius.github.io/ncsdk/ncs.html

[20] S. Haykin and B. Kosko, *GradientBased Learning Applied to Document Recognition*, 2001, pp. 306–351.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] M. Cardim Ferreira Lima, M. E. Damascena de Almeida Leandro, C. Valero, L. C. Pereira Coronel, and C. O. Gonçalves Bazzo, "Automatic detection and monitoring of insect pests—a review," *Agriculture*, vol. 10, no. 5, 2020.

[23] S.-J. Hong, S.-Y. Kim, E. Kim, C.-H. Lee, J.-S. Lee, D.-S. Lee, J. Bang, and G. Kim, "Moth detection from pheromone trap images using deep learning object detectors," *Agriculture*, vol. 10, no. 5, 2020.

[24] A. Rady, N. Ekramirad, A. Adedeji, M. Li, and R. Alimardani, "Hyperspectral imaging for detection of codling moth infestation in goldrush apples," *Postharvest Biology and Technology*, vol. 129, pp. 37 – 44, 2017.

[25] 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, 2018.

[26] N. T. Nam and P. D. Hung, "Pest detection on traps using deep convolutional neural networks," in *Proceedings of the 2018 International Conference on Control and Computer Vision*, ser. ICCCV '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 33–38.

[27] 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.

[28] R. Piyare, A. L. Murphy, P. Tosato, and D. Brunelli, "Plug into a plant: Using a plant microbial fuel cell and a wake-up radio for an energy neutral sensing system," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2017, pp. 18–25.

[29] A. Rodriguez Arreola, D. Balsamo, A. K. Das, A. S. Weddell, D. Brunelli, B. M. Al-Hashimi, and G. V. Merrett, "Approaches to transient computing for energy harvesting systems: A quantitative evaluation," in *Proceedings of the 3rd International Workshop on Energy Harvesting Energy Neutral Sensing Systems*, ser. ENSsys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 3–8.

[30] 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, 2016.