

TinyML Smart Sensor for Energy Saving in Internet of Things Precision Agriculture platform

1st Chollet Nicolas
ECE, LISV
Paris, France
nicolas.chollet@ece.fr

2nd Bouchemal Naila
ECE
Paris, France
naila.bouchemal@ece.fr

3rd Ramdane-Cherif Amar
LISV
Versailles, France
amar.ramdane-cherif@uvsq.fr

Abstract—Smart agriculture researchers bring numerous tools and prospects to the farm ecosystem to improve its productivity and, mainly, its sustainability. Artificial Intelligence (AI) is widely used in precision agriculture as Internet of Things (IoT) technologies have brought a huge volume of data to exploit to provide useful insights for farmers such as weather prediction, pest development detection, or harvest time estimation. AI algorithms are mostly executed in the cloud due to their inherent computing constraints, thus requiring the different sensors to offload their data to the appropriate server. Depending on the amount and volume of data exchanged, the need for computer offloading may induce privacy, security, and latency issues in addition to weighting on the sensor's battery consumption as wireless transmission methods have a high-energy demand. To overcome this difficulty, recent research has tried to bring AI computation closer to the end device with edge or fog computing and more recently with the Tiny Machine Learning (TinyML) paradigm that aims to embed the AI algorithm directly into the sensor's microcontroller. In that context, this paper proposes a prototype of smart sensor capable of detecting fruits presence with TinyML. We then study the energy consumption of our system in different IoT scenarios.

Index Terms—Agriculture, AI (Artificial Intelligence), IoT (Internet of Things), LoRaWAN, Smart Farming, TinyML

I. INTRODUCTION

It is estimated that the world population will increase to reach approximately 9 billion by 2050. The Food and Agriculture Organization (FAO) of the United Nations, therefore, estimates that by then, food production must increase by around 60 % if we want to ensure global food security [1]. To answer this raising concern, Smart Farming (SF) technologies, also called Precision Agriculture (PA) are reshaping the agricultural practices and industry to make them more productive and sustainable [2]. SF tools gather information and communication technologies (ICT) such as Artificial Intelligence, IoT (Internet of Things) Platforms, and Robotics to provide sustainable modern solutions. One of the fields that could bring significant progress to agriculture is the use of AI-powered computer vision analysis to evaluate the crop's growth process or detect unwanted situations such as pest development or weed multiplication. Such AI algorithms rely heavily on the cloud due to their need for heavy computing capacity. This asks for adapted network architectures and raise concern in term of privacy, security, and latency but moreover increase the need for energy as heavy

data offloading call for a larger data throughput [3]. However, battery lifetime is a crucial parameter for agricultural purposes as sensors can be spread over long distances without access to electricity [4]. To overcome those issues, researchers have looked into bringing the computation process closer to the end device that collects the data with fog or edge computing methods to avoid unnecessary communications. However, more recent research on embedded AI has shown that the Tiny ML paradigm now allows micro-controller with small computing capabilities to perform AI directly on the device [5]. In this context future agricultural sensors could perform AI inference directly on the device and only communicates the result as small messages through energy-efficient Low power networks (LPWAN) such as LoRaWAN.

We propose in this paper a prototype of a smart intelligent sensor for fruit presence detection in the context of the smart farm. This proposal is based on the development of embedded vision AI algorithms with Tiny ML. This will allow the farmer to know when and where fruits need attention (for example when to harvest them or when to apply fertilizer) thanks to communication between the AI sensor and the Smart Farm environment using LoRaWAN. An energy analysis of the system in different network scenarios is then conducted to validate our hypothesis that TinyML usage can improve battery lifetime.

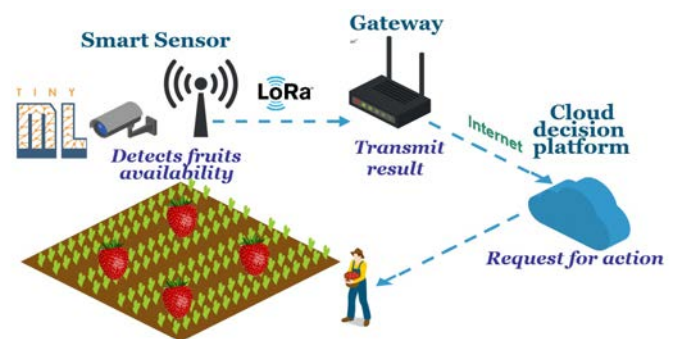


Fig. 1. Overall Architecture

II. STATE OF THE ART

Researches on Tiny ML are relatively new and could be the answer to a wide number of applications. In the Agricultural

domain, the potential applications are vast such as livestock management or insect detection [6], but SF implementations are still limited and we assume that they will be rising in the future. Most research concentrate on the industrial domain, a good survey on recent advancements in Tiny ML has been made by authors in [7]. For other practical implementations, TinyML has been used in multiple scenarios such as adaptive traffic Control [8] or wildlife conservation [9]. TinyML also offers multiple advantages over Fog, Edge, or Cloud computing including improvement in terms of privacy, security, latency, and energy as addressed by the authors in [10].

III. ARCHITECTURE PROPOSAL

A. Scenario

We propose a battery-powered sensor with a camera placed in front of a fruit field. The sensor uses TinyML algorithm to infer the number of fruits. Then, the sensor sends the number to a decision platform through a LoRaWAN network to extend the battery lifetime. This number is then analyzed and a decision is made regarding the necessary action to take, for example, harvesting if enough fruits are present or fertilizer application if fruits are too small or absent. The necessary actions are then communicated to the performers as seen in Figure 1. Our architecture is made up of four main components:

1) **Smart sensor:** In charge of performing the TinyML, it is a microcontroller equipped with a low-resolution camera and a LoRaWAN communication module. We choose LoRaWAN as it is a widely used LPWAN protocol in agricultural Wireless Sensor Network (WSN) as discussed by the authors in [4].

2) **Gateway:** The gateway is in charge of the link between the local LoRaWAN network and the Internet.

3) **Cloud decision platform:** The information received from the sensor is stored in a database. The system then decides regarding other parameters such as time, weather, or farmer's occupation, to ask for an action.

4) **Performer:** The decided action is then communicated to the performer for example a farmer or a robot.

B. Environment development

In order to implement our proposal, we choose the following modules: For the smart sensor, an Arduino Portenta H7 microcontroller with a Lora Vision shield is used. Its 32-bit architecture and low power consumption abilities make it an adapted choice for TinyML algorithms. Moreover, the Arduino ecosystem facilitates the implementation of complex algorithms into microcontrollers that should help the replicability and comparison of our work to other researchers. To create the TinyML algorithm, we used Edge Impulse [11], the development platform based on tensor flow lite. Finally, to communicate with Lora we use a Laird RG1868 gateway and The Thing Network (TTN) environment to store our data online. TTN is a LoRaWAN network server, built on an open-source core that allows users to build and manage LoRaWAN networks easily.

C. Phases

In this section, we describe the end-to-end phases implemented in our prototype.

1) **Phase 1: Data collection:** Collecting data from real devices is the first step to train the model. Recent TinyML development is limited to performing on-device inference and cannot do on-device learning. The creation process of a TinyML model is presented in figure 2. It consists of first gathering data and then pre-train the model independently from the device and afterward deploying it to the hardware and finally testing. To collect the data we use the Edge impulse tool to directly gather pictures from the Arduino Portenta 7. For our experimentation, we trained our model to detect strawberries by collecting 100 pictures containing 0 to 10 instances of the fruit.

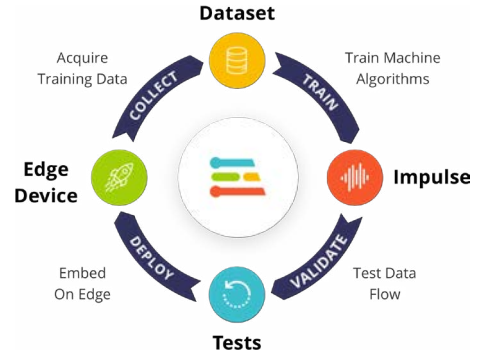


Fig. 2. TinyML algorithm building process with Edge Impulse

2) **Phase 2: TinyML model training :** To perform our fruit detection we use Edge Impulse FOMO (Faster Objects, More Objects) method [11]. It is a novel machine-learning algorithm that brings object detection to highly constrained devices. It allows the device to count objects, find the location of objects in an image, and track multiple objects in real-time using up to 30x less processing power and memory than other similar dedicated algorithms such as MobileNet SSD or YOLOv5 [12].

3) **Phase 3: On-device inferring:** Once the model is deployed on the Hardware, it can detect how many fruits are in front of it. To save energy, the inference process should be performed only for the minimum amount of time so that the microcontroller can stay in sleep mode otherwise.

4) **Phase 4: Results transmission:** After we get the fruit number, we use LoRaWAN protocol to communicate it to a cloud decision platform. This result could be later processed by the hypothetical decision system of our scenario to ask for harvesting.

IV. EXPERIMENTATION

A. Model performance Evaluation

In order to test our solution, the algorithm is implemented directly on the device. For our dataset of 100 images, the accuracy of the FOMO algorithm reaches 90.2% according to Edge

Impulse testing tools and similar accuracy was obtained in real-life tests. The results show that the estimated peak RAM usage is 243,9 kb and the firmware size is 77,5 kb, respectively 24.39 % and 3.88 % of Arduino's capacity, leaving space for some improvement opportunities. The average inference time that will be later used to calculate the energy consumption of the system is 148 ms. You can find the sources on GitHub to reproduce our experiment [13].

B. Energy consumption evaluation

To show the energy effectiveness of our proposal, several simulations were performed using OMNET++ simulator and INET Framework [14] on a PC running Ubuntu 20.04 with 16GB RAM and an Intel I7 8565U. Three scenarios were considered. The first one, where the TinyML algorithm runs directly on the microcontroller and the inference result is communicated through LoRaWAN. The second one, where the TinyML algorithm also runs directly on the microcontroller, and the result is communicated through WiFi. Finally, the third one, where only the taken picture is sent through WiFi to represent data offload and to validate the impact of TinyML on energy consumption. IoT device energy consumption evaluation is a complex process as discussed in [15]. For each scenario, the device only wakes up from sleep mode once a day. When awake, the device is in full power mode during the image capture, the model inference process, and the data communication. During the wireless communication process, we also add the energy consumption of our network interface regarding the volume of data to be transmitted. Either the short message representing the number of fruits detected in scenarios 1 and 2 (50 bytes in our platform) or the image size to transmit in scenario 3 (20kb). Indeed the volume of data to transmit increase the energy consumption of the system and TinyML allows us to send the minimum amount of data required. In order to perform the simulation, we need to collect the energy consumption characteristics of our hardware. In table I, we gather the values found for the different running modes in the hardware datasheet. Thanks to those data the simulator can estimate the approximate energy needed and therefore the impact of each transmission on the battery lifetime.

TABLE I
POWER CONSUMPTION OF THE SYSTEM

Mode	Current Consumption
Standby	2.95 μA
Run	121 mA
Transmission Lora	21.5 mA
Transmission WiFi	310 mA

Finally, the simulator computes the sensor lifetime expectations for a 2000 mA battery until every scenario runs out of power. The results for each one are presented in figure 3 where the evaluation of the battery level over time can be observed.

It appears that Scenario 1 where TinyML and Lora are used, is the most energy-efficient one as it can last up to

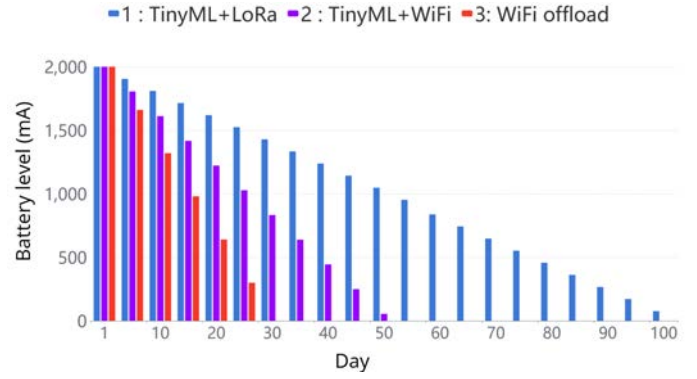


Fig. 3. Evolution of Battery level over time

105 days, this is 3 times longer than Scenario 3. Scenario 2 also show that TinyML can save battery in the case of full WiFi usage as the battery last 1,5 time longer than Scenario 3. The battery lifetime expectation results are regrouped in figure 4. . The results validate the hypothesis that TinyML sensors could increase battery life in such context since it allows us to diminished the need for power-hungry data communication by minimizing the size of the messages to transmit.



Fig. 4. Battery Lifetime of the system in days

V. CONCLUSION

In this paper, TinyML and LoRaWAN were used to propose an energy-efficient model capable of fruit detection to show the capabilities of such technologies in the agricultural domain. Experimentation showed that our model had a 90 % accuracy level and was three times more energy-efficient than a cloud-based model for the same application, opening the way for a new range of computer vision applications in Smart Farming based on battery-powered sensors. Despite promising results, the TinyML paradigm presents some limitations regarding on-device learning capabilities, as the neural network needs to be pre-trained before being embedded into the microcontroller unit. Therefore, the TinyML sensor can not adapt itself to the specific environment it will be deployed in. Later research should be conducted on how to update the model once the sensors are deployed to increase the accuracy after the aquisition, which means performing firmware updates over the air with LPWAN networks for TinyML applications.

- [1] Ranganathan, Janet, et al. "How to sustainably feed 10 billion people by 2050, in 21 charts." (2018).
- [2] Walter, Achim, et al. "Opinion: Smart farming is key to developing sustainable agriculture." *Proceedings of the National Academy of Sciences* 114.24 (2017): 6148-6150.
- [3] Elijah, Olakunle, et al. "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges." *IEEE Internet of things Journal* 5.5 (2018): 3758-3773.
- [4] Jawad, Haider Mahmood, et al. "Energy-efficient wireless sensor networks for precision agriculture: A review." *Sensors* 17.8 (2017): 1781
- [5] Sanchez-Iborra, Ramon, and Antonio F. Skarmeta. "Tinyml-enabled frugal smart objects: Challenges and opportunities." *IEEE Circuits and Systems Magazine* 20.3 (2020): 4-18.
- [6] Zapico, Jorge Luis, Fredrik Ahlgren, and Marco Zenaro. "Insect biodiversity in agriculture using IoT: opportunities and needs for further research." *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021.
- [7] Ray, Partha Pratim. "A review on TinyML: State-of-the-art and prospects." *Journal of King Saud University-Computer and Information Sciences* (2021).
- [8] Roshan, A. Navaas, et al. "Adaptive Traffic Control With TinyML." *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2021.
- [9] Olsson, S. and Efforts, T., 2022. TinyML and IoT for Conservation Efforts. [online] Hackster.io. Available at: <https://www.hackster.io/saraolsson4s/tinyml-and-iot-for-conservation-efforts-dd34db> .
- [10] Aslanpour, Mohammad S., Sukhpal Singh Gill, and Adel N. Toosi. "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research." *Internet of Things* 12 (2020): 100273..
- [11] Moreau, L. and Kelcey, M., 2022. Announcing FOMO (Faster Objects, More Objects). [online] Edge Impulse. Available at: <https://www.edgeimpulse.com/blog/announcing-fomo-faster-objects-more-objects>
- [12] Aadithya, V., et al. "Comparative Study Between MobilNet Face-Mask Detector and YOLOv3 Face-Mask Detector." *Sustainable Communication Networks and Application*. Springer, Singapore, 2022. 801-809.
- [13] Chollet, N., 2022. GitHub - Portenta_Fragola. [online] GitHub. Available at: https://github.com/NicolasChollet51/Portenta_Fragola
- [14] Mészáros, Levente, Andras Varga, and Michael Kirsche. "Inet framework." *Recent Advances in Network Simulation*. Springer, Cham, 2019. 55-106.
- [15] Morin, Elodie, et al. "Comparison of the device lifetime in wireless networks for the internet of things." *IEEE*