WideVision: A Low-Power, Multi-Protocol Wireless Vision Platform for Distributed Surveillance

Moritz Scherer[†], Fabian Sidler[†], Michael Rogenmoser[†], Michele Magno[†], Luca Benini[†‡]
[†]Dept. of Information Technology and Electrical Engineering, ETH Zürich, Switzerland
[‡]Dept. of Electrical, Electronic and Information Engineering, University of Bologna, Italy

***Abstract*— The trend in Internet of Things research points toward performing increasingly compute-intensive data analysis tasks on embedded sensor nodes, rather than server centers. Exploiting the technological advances in both energy efficiency, and Tiny Machine Learning algorithms and methods, an increasing number of recognition and classification tasks can be performed by small, low-power, wireless sensor nodes. This paper presents WideVision, a wireless, wide-area sensing platform capable of performing on-board person detection with power requirements in the mW range. The WideVision platform integrates seamlessly into the Internet of Things, by coupling a dedicated multi-radio platform, including a LoRa interface, enabling medium- and long-range communication, with a novel parallel RISC-V microcontroller. We evaluate the proposed platform with the GAP8 microcontroller, which includes an 8-core RISC-V cluster, and greyscale camera to perform person detection by training and deploying an advanced, quantized neural network, achieving a statistical accuracy 84.5% for a 5-person detection task with a latency of only 182 ms. Experimental results demonstrate that the WideVision sensor node platform while performing inference at a rate of one image per minute on-board, is capable of lasting 300 days on a 2400 mAh Li-ion battery, and 65 days when evaluating one image per 10 seconds while providing effective surveillance of its perimeter.**

I. Introduction

In recent years, Internet of Things (IoT) devices have become more and more intelligent and pervasive in all areas of everyday life [1]. Enabled by low-power sensors and a high degree of technological integration, IoT devices cover a wide range of applications. The first generations of IoT devices stream the raw data they acquire to a centralized server, where useful information is extracted. This paradigm of cloud-centric computing has significant disadvantages: Since raw data is transmitted, the lion's share of power consumption is consumed by the radio interface of the system. Further, the latency of the system is dictated by the round-trip time between the node and the cloud. Especially with sensor networks that produce a large amount of rapidly changing data and require a short reaction latency, edge computing improves performance, battery lifetime, and cost. [2].

In edge computing, sensor nodes apply advanced algorithms, often from the domain of machine learning, to extract useful information from the raw data they acquire. Edge computing not only minimizes the amount of data sensor nodes have to transmit via radio, which significantly improves the latency of the system [3] . The new generation of edge computing-focused Microcontrollers (MCUs) often features on-chip memory in the order of hundreds of kilobytes and can process sub-word data efficiently by exploiting specialized Instruction Set Architecture (ISA) extensions. Some modern MCUs even feature multi-core compute clusters, further decreasing the latency of Deep Neural Network (DNN) inference [4], [5]. Following the advancement of specialized MCUs, deployment of complex DNNs has become feasible without compromising the battery life of their host system, giving rise to the field of TinyML [5], [6].

Adapting surveillance tasks to this emerging paradigm of edge computing is quite challenging. Surveillance requires extensive area coverage by sensors, and accurate detection and classification to be effective, posing significant challenges for embedded systems [7], [8]. A distributed approach where many wide-area sensor nodes cover partial perimeters in combination with a few long-range nodes often works well in practice to adapt to occluded areas [7], [9]. Ideally, these nodes are battery-powered and able to communicate events quickly, to minimize both the required maintenance and reaction time.

This work proposes a novel system that tackles the challenges distributed wide-area and long-range sensor systems face, by designing and implementing a multi-radio platform. We further integrate this platform into a low-power surveillance system, using an off-the-shelf RISC-V multi-core MCU board and camera. By training and deploying a Convolutional Neural Network (CNN) we demonstrate that our platform achieves state-of-the-art energy efficiency for vision-based surveillance applications.

Experimental results with in-field measurements show that our solution has a connectivity range of several kilometers using the LoRa interface while sending few bytes of processed information such as the detection of a person. The sensor node detects and counts people with an accuracy of $84.5\,\%$ and an inference latency of $182\,\mathrm{ms}$. Experimental results show that the proposed solution can last $65\,\mathrm{d}$ in the field using an off-the-shelf $2400\,\mathrm{mA\,h}$ battery pack, enabling a long-lasting, decentralized sensing network.

II. Related Work

In the field of smart sensor nodes, the topics of embedded Machine Learning and edge computing have seen numerous contributions in the past few years. A wide range of hardware

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

platforms and machine learning algorithms have been studied to enable deployable systems [10].

Distributed and embedded surveillance systems are well-researched, with many contributions focusing on a variety of application scenarios, ranging from agricultural to healthcare monitoring [9], [11]–[15]. While the body of work is fairly extensive, most of the proposed designs are either not optimized for ultra-low power battery-based operation [11], [12], [15] or too inflexible for reliable surveillance, due to strict constraints on their power consumption [13], [14]. Nevertheless, some recent research has investigated low-power, battery-operated distributed surveillance systems or similar concepts.

The authors in [7] propose a Wi-Fi-based wireless multimedia sensor network for IoT applications. To reduce the power consumption by the cameras, which stream their video output via Wi-Fi, the authors propose to use a Long Range (LoRa) network to control the activity of each node. Their solution is shown to reduce the system-level power consumption since power is consumed by fewer nodes at a time. In a fully embedded, battery-operated surveillance setting, however, this solution is largely infeasible, since if a camera is offline, some part of the perimeter is left unsupervised.

Benninger et al. [16] present a sensor node based on the GAP8 microcontroller, combined with two microphones, a camera, a light sensor, and a LoRa radio. By exploiting aggressive power optimization techniques, the authors reduce the average power consumption of the system to around $770\,\mu\mathrm{W}$, assuming one inference is performed per minute. While this system's power consumption is very impressive, it is unable to adapt to different latency and throughput requirements, since the only available radio interface is LoRa. Due to its highly integrated nature, it is also not possible to exchange the sensors or main processor.

The closest work to ours is presented by Polonelli et al. [17]. The authors propose a multi-radio shield featuring LoRa, Wi-Fi, and Bluetooth radio interfaces. While their system demonstrates flexibility and energy efficiency for what concerns the radio interface, the evaluation only considers the radio interface energy and does not consider edge computing aspects as well as deployment constraints and issues. Thus, in this paper, we focus on the tiny machine learning aspect and the long-range wireless interface using LoRa to send processed information to a remote gateway up to several km of distance.

We show on an end-to-end system that our approach reduces the energy cost of surveillance in an embedded setting by moving the computational burden on the sensor board and significantly reducing the energy cost of LoRa radio transmission, leading to state-of-the-art energy efficiency and accuracy in a low-power surveillance system.

III. System Architecture

In a typical sensor network systems application scenario, several sensor nodes communicate with a central gateway, forming a star topology network, as shown in Figure 1. Implementing networking with a star topology saves energy since sensor nodes do not need to route packets. The range over which sensor nodes need to transmit data depends on the application scenario. In the context of highly distributed systems, the required distance might reach kilometer range [8], while other applications might only require tens or hundreds of meters [18]. Similarly, the throughput requirements for applications vary enormously. While some applications require continuous streaming of data, others only require transmitting compressed classification data.

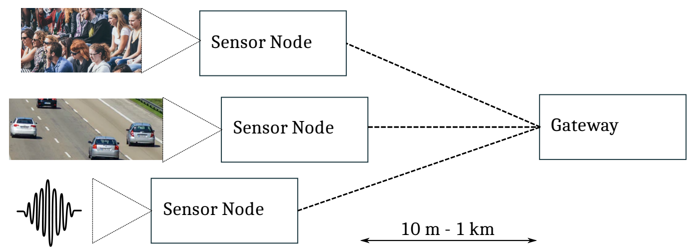

Fig. 1. High-level view of the application scenario. The sensor nodes monitor distinct areas and communicate with the gateway in a star topology network.

To enable a trade-off between range, throughput, latency, and transmission power, as required by different applications, we propose the design of a multi-radio platform, which can equip any off-the-shelf microcontroller with multi-protocol radio support, enabling both the streaming of data over short distances as well as the transmission of compact data packets over large distances. As shown in Figure 2, the sensor node consists of a sensor, a data processing unit, as well as the multi-radio platform. The processing unit extracts useful information and/or compresses the sensor data to reduce the transmitted size. The custom multi-radio platform has a standard Arduino UNO connector and provides wireless connectivity as well as a battery power circuit. It converts the host platform to an IoT capable sensor node supporting LoRa, Wi-Fi and Bluetooth. Each radio interface on the multi-radio platform is designed to enable different operating modes and deployment scenarios: While the LoRa interface allows communication in the range of multiple kilometers, the Bluetooth and Wi-Fi interfaces can be used to trade-off range for transmission rate and decreased latency.

In this work, we evaluate the LoRa radio interface in a distributed surveillance application scenario, with a special focus on minimizing the power consumption during idle periods.

IV. Implementation

In this section, we discuss the proposed multi-radio system and how we extend it by implementing a sensor node architecture for surveillance applications. Our implementation is capable of transmitting highly compressed information extracted from images over long distances, reaching up to $10\,\mathrm{km}$.

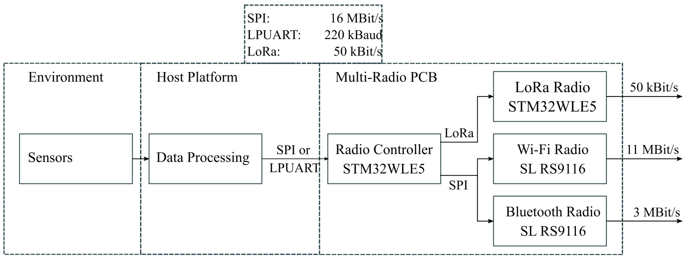

Fig. 2. Architecture of sensor nodes using the proposed multi-radio platform. Data Processing is performed on the host platform, while the multi-radio platform handles transmission of raw or compressed data. Depending on the application, the multi-radio receives data over SPI or UART.

A. MultiRadio Platform

The multi-radio platform is designed to both minimize transmission energy, as well as idle power consumption. We achieve this by implementing a dedicated DC/DC buck converter, which can be used to fully power gate the components on the PCB. Figure 3 shows a diagram of the PCB's schematics.

To power the sensor node, a 1S Lithium-ion battery is required. The battery is protected and charged by a dedicated circuit built around a Texas Instruments BQ24232HARGTR Integrated Circuit (IC). The multi-radio platform provides the $3.7\,\mathrm{V}$ supply of the Li-Ion battery directly to the host platform, while the radio interfaces and the radio controller are supplied with $2\,\mathrm{V}$ from a MAXM38643AEMB+ buck converter by Maxim Integrated. To reduce power consumption in idle mode, The buck converter can be completely powered off by the host platform.

This results in two distinct power domains on the multi-radio platform, one of which is powered by the battery voltage which is always on, and a switched $2\,\mathrm{V}$ power domain. To allow communication between any host platform and the $2\,\mathrm{V}$ logic on the multi-radio platform, logic level shifters are integrated on the multi-radio platform.

The processing unit on the host platform communicates over Serial Peripheral Interface (SPI) with the radio controller, an ARM Cortex M4 processor by ST microelectronics. We chose the STM32WLE5JC, since it integrates a LoRa radio with two transmitting power options and its power consumption of $72\,\mu\mathrm{A}\,\mathrm{MHz}^{-1}$ is one of the lowest found in commercial off-the-shelf microcontrollers with integrated LoRa radio interface.

Additionally, a Silicon Labs RS9116-CC1 radio module is integrated for Wi-Fi and Bluetooth connectivity. The RS9116 is connected with the radio controller over an SPI interface, as shown in Figure 3. The radio controller is used to handle the protocol stack of all radio interfaces. This allows the sensor node to be fully decoupled in terms of processing and data transmission; The host microcontroller can be fully power-gated while the radio controller is still able to communicate with the gateway. Vice-versa, the radio controller can be fully power-gated when no communication is required. The setup with the STM32WLE5 as a single interface to the host platform and the incorporation of logic level converters results in an energy-efficient and portable multi-radio platform.

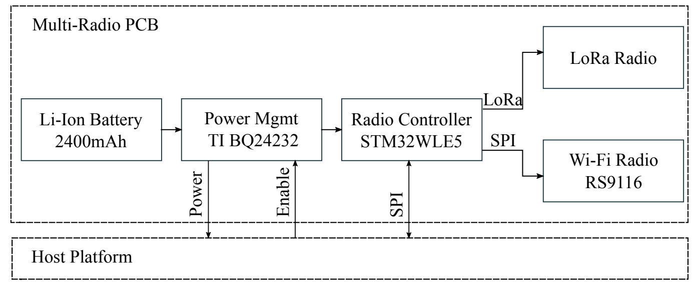

Fig. 3. Block diagram of the proposed multi-radio platform. The platform features a battery management circuit, used to power the host platform, and a DC/DC buck converter, which powers the Radio circuitry and enables the host platform to power off the radio controller and its interfaces.

B. Sensor Node

We implemented a full sensor node using the GAPuino evaluation board. A GAP8 development platform with an integrated camera connector is used as the host platform for sensor data processing. We use the Himax HM01B0 embedded camera, a state-of-the-art embedded low-power QVGA greyscale camera. The custom multi-radio platform is used to provide the sensor node with Long Range Wide Area Network (LoRaWAN) connectivity and power. In this setup, the GAPuino is the host platform for the multi-radio platform. A schematic of this setup is depicted in Figure 4 and a picture of the complete sensor node is shown in Figure 5.

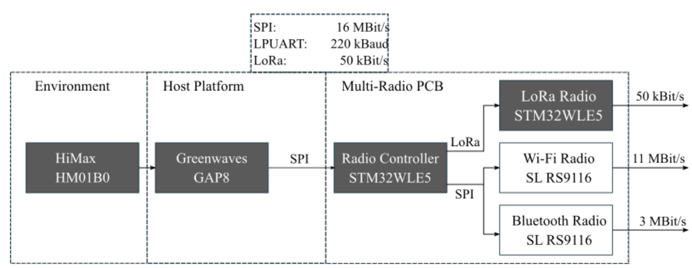

Fig. 4. High level schematic of a sensor node used for evaluation. An embedded grayscale camera is used as an image sensor and the GAPuino's octa-core cluster is used for machine-learning image classification.

C. Neural Network

One of the contributions of this work is the training, quantization, and deployment of a CNN used for person detection on the host platform. We propose a model that minimizes the memory and computational resource usage of the GAP8 microcontroller, which typically poses the greatest challenge for deployment of neural networks on small embedded devices, while still offering the required accuracy for surveillance applications. The GAPuino board used in this implementation is a development board designed around the GAP8 microcontroller,

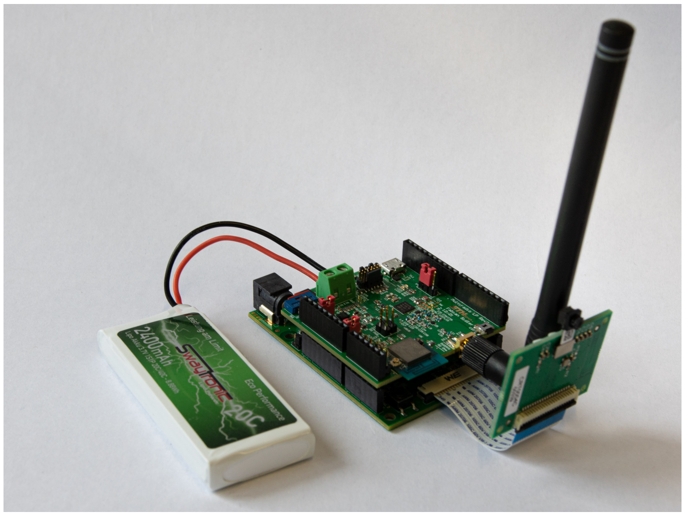

Fig. 5. Picture of the assembled surveillance sensor node, consisting of an off-the-shelf Li-Ion battery, the GAPuino, the Himax camera, as well as the custom multi-radio platform.

and octa-core RISC-V microcontroller featuring $512\,\mathrm{kB}$ of on-chip memory, based on the PULP SoC [19]. Besides this microcontroller, the GAPuino also features $16\,\mathrm{MB}$ of external HyperRAM Dynamic Random Access Memory (DRAM) memory, which is used for storage of model weights and intermediate feature maps in large models.

a) Model architecture & Training: The proposed model architecture is a modified version of MobileNet-v1 [20]. Similar to the original architecture, we use 3×3 Depthwise-separable convolutions with strides to reduce the feature map dimensions and reduce the computational and memory requirements compared to dense convolutions. To ease quantization, each convolution is followed by a batch normalization layer [21] and a ReLU activation [22]. Table I lists the model parameters. The model classifies each input image and outputs the number of people in the field of view in 5 classes, including classes for 0-3 people and a class for more people. We used the Adam optimizer [23] with an initial learning rate of 10^{-3} and a learning rate scheduler that reduces the learning rate by $3\times$ if the validation loss has not improved for 10 epochs.

b) Dataset & Preprocessing: The use of large, sensor-specific datasets is an important factor for the successful deployment of machine learning models to embedded systems. Since such datasets are rarely openly accessible, and to help with the comparability of our solution, we used the COCO dataset [24] to train our model. Since the COCO dataset uses high-resolution images annotated with bounding boxes that locate the position of objects, including people, we used a custom pre-processing pipeline to rescale the images and annotate them with the classification labels required by our network. We first apply a random rescaling of the image, randomly crop a patch of size 320×240, apply RandAugment [25] with parameters $N = 2$, $M = 14$ and convert the image to greyscale.

c) Quantization & Deployment: To execute the model on the GAP8 platform, it needs to be quantized, such that it can be executed using only integer arithmetic since the GAP8 does not have a Floating Point Unit (FPU). Quantizing the model's weights and activations to $8\,\mathrm{bit}$ further helps to drastically reduce the memory requirements for storing model weights and feature maps as well as the compute requirements since the ISA extensions of the RI5CY cores inside the GAP8 enable parallel Single Instruction Multiple Data (SIMD) processing. We used the Quantlib[1] framework to quantize both the weights and activations of the network to $8\,\mathrm{bit}$, and the Dory framework [5] to deploy it on the GAP8 microcontroller.

d) Evaluation: The model has a total size of $90.5\,\mathrm{kB}$ and requires $103.6\,\mathrm{MOps}$ per inference. Deploying it on GAP8 and running on all 8 cores in the cluster, the inference takes a total of $18.2\,\mathrm{MCycles}$, achieving an average compute intensity of $5.45\,\mathrm{Op/Cycle}$. On the GAP8 running at $100\,\mathrm{MHz}$, inference takes $182\,\mathrm{ms}$. The model achieves an average accuracy of 84 % on the validation dataset.

TABLE I
MODEL ARCHITECTURE OF THE PROPOSED NEURAL NETWORK

Layer	**Input**	**Stride**	**Channels**
Dense 2D Convolution	320×240	2	32
Separable 2D Convolution	160×120	2	48
Separable 2D Convolution	80×40	1	64
Separable 2D Convolution	80×40	2	64
Separable 2D Convolution	40×20	1	64
Separable 2D Convolution	40×20	2	64
Separable 2D Convolution	20×10	2	96
Separable 2D Convolution	10×5	2	96
Separable 2D Convolution	5×3	2	128
Linear Layer	768	-	5

V. Power and Energy Consumption Analysis

To test the functionality and measure the performance of our proposed platform, we use the sensor node system described in Section IV, consisting of the multi-radio platform and the GAPuino microcontroller board. We measure the idle power consumption, as well as the total energy cost of taking an image, processing it using the neural network, and transmitting the classification results with the LoRa radio interface. We further calculate the energy cost of transmitting a raw image, to compare our approach. Finally, we estimate the overall system lifetime using the attached battery depending on the frequency with which images are processed.

To evaluate the contribution of each device individually, the power consumption of the multi-radio platform, GAP8 processor, and HiMAX camera were measured separately and added afterward.

[1]https://github.com/pulp-platform/quantlib

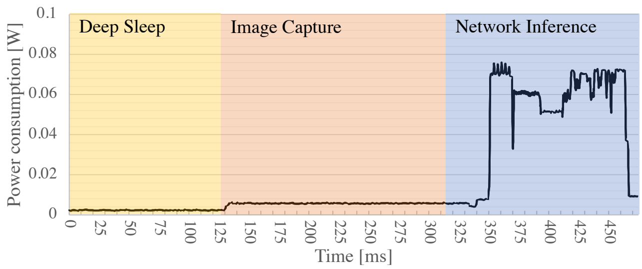

Fig. 6. Power measurement of image capture and neural network inference on the GAPuino board.

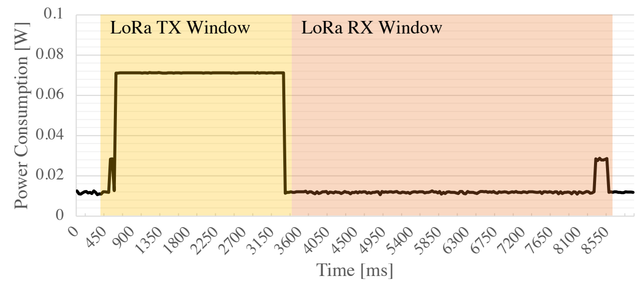

Fig. 7. Power measurement of the multi-radio PCB during transmission via the LoRa interface at DR0, at 18 dBm transmission power.

A. GAPuino Evaluation Setup

To evaluate the efficiency of the host processor and camera sensor, we measure the power consumption of the GAPuino board, as used in our prototype system. Since the GAPuino is equipped with non-optimal Low Dropout Regulators (LDOs) that convert the battery voltage of $3.7\,\mathrm{V}$ to the operation voltage of $1.2\,\mathrm{V}$, leading to an efficiency loss of at least 66 %, we only measure the power consumption of the GAP8 processor and the Himax camera, both while idle as well as during inference and image acquisition. We use the measured power consumption of the processor and camera to estimate the energy cost of acquiring, analyzing, and transferring an image.

B. MultiRadio Evaluation Setup

To measure the power consumption during startup, LoRaWAN joining sequence, transmission, and reception of the multi-radio platform the Printed Circuit Board (PCB) was powered using a Keysight N6715C DC power analyzer. The power analyzer provides $3.7\,\mathrm{V}$ to the multi-radio platform over the Li-ion battery connector and monitors the current consumption. In our measurement setup, the multi-radio platform transmits a LoRa package with $16\,\mathrm{B}$ of payload to the gateway using DR0 and DR5, respectively[2] (SF8, $125\,\mathrm{kHz}$ bandwidth and $3125\,\mathrm{bit\,s^{-1}}$) at $18\,\mathrm{dBm}$ output power. This accurately models the payload size of our implementation, since we only transmit the device ID, a timestamp, and the inference result, which total $6\,\mathrm{B}$. However, due to the AES encryption used in the LoRaWAN protocol, the minimum payload size is $16\,\mathrm{B}$. Figure 7 shows the power trace of the multi-radio PCB for a transmission of a classification result with DR0. Using these measurements, we consider the packet TX window and one RX window as the basis for our energy consumption calculations.

C. Power Measurement Results

To estimate the lifetime of the sensor node, we profiled both the idle power consumption of the sensor node and the multi-radio platform, as well as the energy cost of acquiring an image, running the neural network, and transmitting the classification result with the LoRa interface. With the processor in deep-sleep mode, the camera turned off, and the real-time clock turned on, we measured a power consumption of $156\,\mu\mathrm{W}$ on the GAPuino board. With the DC/DC converter on the multi-radio platform turned off, we measured a power consumption of $270\,\mu\mathrm{W}$ from the battery. Hence, while in deep sleep, waiting to acquire and process another image, the platform consumes $426\,\mu\mathrm{W}$. The average power consumption of the GAP8 processor during inference is $51\,\mathrm{mW}$ at $100\,\mathrm{MHz}$, leading to a total energy cost per inference of $9.5\,\mathrm{mJ}$. The acquisition and transfer of an image require around $1\,\mathrm{mJ}$. Figure 6 shows the power trace of one image acquisition and neural network inference on the GAPuino board. The transmission of a classification result at $18\,\mathrm{dBm}$ using the DR0 mode of the LoRa interface consumes around $240\,\mathrm{mJ}$ over $8.5\,\mathrm{s}$. In DR5 mode, the transmission time is reduced to $5.2\,\mathrm{s}$, resulting in an energy cost of $61.4\,\mathrm{mJ}$. When using DR0 for data transmission, the overall energy cost of image acquisition, inference, and transmission comes up to $250.5\,\mathrm{mJ}$, in DR5 mode the overall energy cost comes up to $72\,\mathrm{mJ}$.

D. Lifetime Estimation

Using the measurements presented in the previous section, we model the energy consumption by computing the average power consumption depending on the ratio between deep-sleep and active mode. We further calculate the energy in the fully charged battery to be $35\,964\,\mathrm{J}$, assuming a capacity of $2400\,\mathrm{mAh}$ at $3.7\,\mathrm{V}$. Assuming a high image processing frequency of 1 image per 10 seconds and every inference triggering a transmission, our platform can last at least $16.7\,\mathrm{d}$. Assuming one image is processed every 60 seconds, our platform can last for at least $92\,\mathrm{d}$. When operating on a shorter range, where transmissions using the DR5 mode are possible, these lifetime estimates increase to $65\,\mathrm{d}$ and $300\,\mathrm{d}$, respectively.

E. Comparison With Related Work

When directly comparing our results to [16], we find that our estimate for the system's lifetime is $100\,\mathrm{d}$, or $25\,\%$ lower. This can be explained by two aspects of our implementation and evaluation. Firstly, our neural network is more complex,

[2]https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

which leads to higher energy costs for inference. Secondly, the authors in [16] do not consider the energy cost of receiving an answer from the gateway during the RX window of LoRa transmissions, which neglects a large amount of the real energy cost of running a practical radio system.

We further compare with the work by Polonelli et al. [17], which reports energy per Bit numbers for transmission with LoRa. In our evaluation, we find that our transmission energy cost per Bit in DR0 is about $5\,\mathrm{mJ}$, $44\,\%$ lower than their solution, at an equal LoRa TX power of $20\,\mathrm{dBm}$. The energy cost per Bit figure in [17] adds up the average power of the antenna, the SemTech 1276 chip, and communication between their microcontroller and SemTech chip. Since our solution combines the LoRa frontend and microcontroller, we avoid costly inter-chip communication, which explains the increased efficiency of our solution.

VI. Conclusion

This paper presents a multi-radio platform for IoT applications, that aims to easily convert any microprocessor into an IoT capable sensor node with a battery power circuit. The multi-radio platform is evaluated in a wide-area surveillance application, using a GAPuino development board featuring the GAP8 low-power RISC-V multicore processor. A CNN for person recognition was trained and deployed on the GAPuino. The power consumption of both the machine learning and the data transmission over LoRaWAN were measured, and estimates for the sensor lifetime were calculated, based on measurements of the system's idle and active power. We show that our system can last $300\,\mathrm{d}$ on a single battery charge while performing accurate surveillance in the field.

Acknowledgement

The authors would like to thank *armasuisse Science & Technology* for funding this research.

References

[1] S. Kumar *et al.*, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *Journal of Big Data*, vol. 6, no. 1, p. 111, Dec. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0268-2

[2] H. Li *et al.*, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan. 2018, conference Name: IEEE Network.

[3] L. Corneo *et al.*, "(How Much) Can Edge Computing Change Network Latency?" in *2021 IFIP Networking Conference (IFIP Networking)*, Jun. 2021, pp. 1–9, iSSN: 1861-2288.

[4] E. Flamand *et al.*, "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)*, Jul. 2018, pp. 1–4, iSSN: 2160-052X.

[5] A. Burrello *et al.*, "DORY: Automatic End-to-End Deployment of Real-World DNNs on Low-Cost IoT MCUs," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1253–1268, Aug. 2021, arXiv:2008.07127 [cs]. [Online]. Available: http://arxiv.org/abs/2008.07127

[6] R. David *et al.*, "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, Mar. 2021. [Online]. Available: https://proceedings.mlsys.org/paper/2021/hash/d2ddea18f00665ce8623e36bd4e3c7c5-Abstract.html

[7] E. Dilla Diratie *et al.*, "Hybrid Internet of Things Network for Energy Efficient Video Surveillance," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, Jun. 2020, pp. 1–6.

[8] Z. Zinonos *et al.*, "Grape Leaf Diseases Identification System Using Convolutional Neural Networks and LoRa Technology," *IEEE Access*, vol. 10, pp. 122–133, 2022, conference Name: IEEE Access.

[9] I. Ledakis *et al.*, "Adaptive Edge and Fog Computing Paradigm for Wide Area Video and Audio Surveillance," in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, Jul. 2018, pp. 1–5.

[10] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157816300799

[11] Z. Baird *et al.*, "A CNN-LSTM Network for Augmenting Target Detection in Real Maritime Wide Area Surveillance Radar Data," *IEEE Access*, vol. 8, pp. 179 281–179 294, 2020, conference Name: IEEE Access.

[12] F.-J. Romero-Paisano *et al.*, "Accurate multi-target surveillance system over wide areas," in *2017 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2017, pp. 1–6, iSSN: 2153-0742.

[13] H. Desai *et al.*, "Camaroptera: A Long-Range Image Sensor with Local Inference for Remote Sensing Applications," *ACM Transactions on Embedded Computing Systems*, Jan. 2022, just Accepted. [Online]. Available: https://doi.org/10.1145/3510850

[14] M. Giordano *et al.*, "A Battery-Free Long-Range Wireless Smart Camera for Face Recognition," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 594–595. [Online]. Available: https://doi.org/10.1145/3485730.3493367

[15] M. Brown *et al.*, "Multi-Modal Detection Fusion on a Mobile UGV for Wide-Area, Long-Range Surveillance," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2019, pp. 1905–1913, iSSN: 1550-5790.

[16] S. Benninger *et al.*, "EdgeEye: A Long-Range Energy-Efficient Vision Node For Long-Term Edge Computing," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, Oct. 2019, pp. 1–8.

[17] T. Polonelli *et al.*, "A multi-protocol system for configurable data streaming on IoT healthcare devices," in *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*, Jun. 2019, pp. 112–117.

[18] K. S. Dheman *et al.*, "ImpediSense: A long lasting wireless wearable bio-impedance sensor node," *undefined*, 2021. [Online]. Available: https://www.semanticscholar.org/paper/ImpediSense%3A-A-long-lasting-wireless-wearable-node-Dheman-Mayer/bb5798308c340744ee37e9b4da07c78b7936b130

[19] D. Rossi *et al.*, "PULP: A parallel ultra low power platform for next generation IoT applications," in *2015 IEEE Hot Chips 27 Symposium (HCS)*. IEEE, Aug. 2015, pp. 1–39.

[20] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv, Tech. Rep. arXiv:1704.04861, Apr. 2017, arXiv:1704.04861 [cs]. [Online]. Available: http://arxiv.org/abs/1704.04861

[21] S. Ioffe *et al.*, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv, Tech. Rep. arXiv:1502.03167, Mar. 2015, arXiv:1502.03167 [cs]. [Online]. Available: http://arxiv.org/abs/1502.03167

[22] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," arXiv, Tech. Rep. arXiv:1803.08375, Feb. 2019, arXiv:1803.08375 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1803.08375

[23] D. P. Kingma *et al.*, "Adam: A Method for Stochastic Optimization," arXiv, Tech. Rep. arXiv:1412.6980, Jan. 2017, arXiv:1412.6980 [cs]. [Online]. Available: http://arxiv.org/abs/1412.6980

[24] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," arXiv, Tech. Rep. arXiv:1405.0312, Feb. 2015, arXiv:1405.0312 [cs]. [Online]. Available: http://arxiv.org/abs/1405.0312

[25] E. D. Cubuk *et al.*, "RandAugment: Practical automated data augmentation with a reduced search space," arXiv, Tech. Rep. arXiv:1909.13719, Nov. 2019, arXiv:1909.13719 [cs]. [Online]. Available: http://arxiv.org/abs/1909.13719