Accelerated Visual Context Classification on a Low-Power Smartwatch

Accelerated Visual Context Classification on a Low-Power Smartwatch

Francesco Conti, *Student Member, IEEE,* Daniele Palossi, Renzo Andri, Michele Magno, *Member, IEEE,* Luca Benini, *Fellow, IEEE,*

***Abstract*—Data produced by wearable sensors is key in contexts such as performance enhancement and training help for sports and fitness, continuous monitoring for aging people and for chronic disease management, and in gaming and entertainment. Unfortunately, wearable devices currently on the market are either incapable of complex functionality or severely impaired by short battery lifetime. In this work, we present a smartwatch platform based on an ultra-low power (ULP) heterogeneous system composed by a TI MSP430 microcontroller, the PULP programmable parallel accelerator and a set of ULP sensors, including a camera. The embedded PULP accelerator enables state-of-the-art context classification based on Convolutional Neural Networks (CNNs) to be applied within a sub-10mW system power envelope. Our methodology enables to reach high accuracy in context classification over 5 classes (up to 84%, with 3 classes over 5 reaching more than 90% accuracy), while consuming 2.2mJ per classification, or an ultra-low energy consumption of less than 91uJ per classification with an accuracy of 64% - 3.2× better than chance. Our results suggest that the proposed heterogeneous platform can provide up to 500× speedup with respect to the MSP430 within a similar power envelope, which would enable complex computer vision algorithms to be executed in highly power-constrained scenarios.**

***Index Terms*—Machine vision, Wearable computers, Low-power electronics, Energy efficiency.**

I. Introduction

THE VAST improvements in device miniaturization and performance due to the continuous advance of Moore's Law, along with the availability of ubiquitous wireless connectivity, are today enabling the development of smaller and smaller devices that can leverage a relatively high amount of processing performance, and at the same time are always on. A fast growing class of such devices is *smart wearables*, where electronics and sensors are tightly coupled with the human body [1]; this paradigm proposes to transform everyday life objects such as wrist watches, necklaces and glasses in "smart" objects that look promising for a plethora of applications, such as sports and fitness, augmented reality and personalized health care; moreover, top-tier hi-tech companies such as Google, Samsung and Apple look at wearable devices as a new high growth segment in the consumer market. Smart wearable devices open up new possibilities in terms of context awareness [2], making all devices more conscious of their environment and therefore more "intelligent". Continued miniaturization and power improvements have eased the construction of a wide variety of wearable multi-sensor systems [3]. In fact, some forecasts preview up to a trillion connected devices, which are going to produce a huge amount of data [2]. Even with so many sensor-rich wearables, however, the sheer amount of data alone will not provide any value, unless it is possible to turn it into actionable, contextualized information. Machine learning technologies are used with great success in many application areas, solving real-world problems in entertainment systems, robotics, health care, and surveillance [4]; they are extremely flexible and can be applied to heterogeneous data. However, due to their massive requirements in terms of memory and computational throughput, these high accuracy techniques are currently considered to be too computationally expensive for the limited capabilities of wearable devices; instead, sensory data is transmitted to servers "in the cloud"[5] at a high cost in terms of latency and transmission energy.

At the same time, one of the main limitations of the current generation of wearable devices is autonomy, due to the limited amount of energy that can be stored in the batteries. Continuous transmission of data is expensive in terms of energy and severely hinders the autonomy of these devices, posing a practical limit to the amount of useful information that a wearable device can send to the cloud for processing. An alternative approach is that of partially performing the processing locally to the wearable node, so that what is sent out via wireless communication is data in a high-level format (such as visual features) and of reduced dimensionality. This is a major challenge for a typical low-power wearable device driven by a low-power microcontroller unit (MCU). Off-the-shelf MCUs are orders of magnitude less powerful than it would be necessary to sustain data classification using state-of-the-art machine learning techniques [4][6]. As a possible solution to this challenge, *parallel programmable accelerators* have been proposed [7][8] as a means to obtain the necessary level of performance while keeping the power envelope controllable. Accelerators for wearable computers need to perform a variety of tasks and algorithms to fuse data coming from several sensor sources. To provide the necessary level of performance and energy efficiency for this class of algorithms, it is necessary to use deeply integrated technologies that come with high engineering and manufacturing costs. As a consequence, accelerators need to be flexible, *i*) to be coupled to many different host devices (e.g. MCUs) and *ii*) to be applied to

The authors are with the Integrated Systems Laboratory at ETH Zurich, Switzerland. Francesco Conti, Michele Magno and Luca Benini are also with the Department of Electrical, Electronic and Information Engineering at the University of Bologna, Italy. This work has been funded by the *MicroLearn: Micropower Deep Learning* Swiss SNF project, and by projects *IcySoC* and *YINS RTD*, evaluated by the Swiss SNF and funded by Nano-Tera.ch with Swiss Confederation financing.

a very wide range of scenarios, enabling cost-efficient economy of scale.

One of the target applications for wearable devices is that of *ego-vision*, i.e. vision using a first person video stream as the primary source of information. Ego-vision enables use cases such as gesture recognition for augmented reality with off-the-shelf smartphones [9] or a Google Glass device [10], sign recognition to assist people with visual impairments [11], eye movements detection, on top of applicative scenarios such as assisted living (fitness, entertainment, etc.) [12], health-care assistance [13], adaptive environments [14], *Internet of Things* ecosystems [15], and advanced human-machine interfaces driven by hand/eye movement. An ego-vision system can be used to achieve a multi-node assisted environment (e.g. house, car, gym, office, etc.) where complex multi-device behavior is triggered by an "intelligent device" always aware of the user's activity [14]. As all of the mentioned scenarios are time-critical applications, fast computation plays an important role for fast "detect and act" capability [12] - onboard computation can provide a definite advantage by minimizing latency, and open the road to these many diverse applications being continuously running directly on-body. More advanced ego-vision applications are in the context of a multi-device system, where several body-coupled sensors interact in real time with IoT devices in the environment. This would enable deeper and smarter context understanding scenarios. However, such a tight interaction necessitates low latency and exchange of relatively small, semantically rich information as opposed to raw sensor data - therefore necessitating a computationally powerful wearable computer. A fast, unintrusive, and low-power "personal hub" device could be the key enabler for such a system.

In this work, we propose a low power platform for wearable computing and ego-vision, based on a heterogeneous system composed by a Texas Instruments MSP430 microcontroller and a ultra-low power parallel accelerator, the PULP3 chip. The system is equipped with ultra-low power sensors: an analog camera, a microphone, acceleration and temperature sensors. We deploy this platform on a wearable smartwatch device. The proposed approach enhances the application scenarios where on-board processing (i.e. without streaming out the sensor data) enables intensive computation to extract complex features. The smartwatch platform forms a challenging environment for vision due to lighting, obstruction and continuous motion; we show that by using one of the algorithms enabled by our platform (a convolutional neural network) it is possible to extract meaningful information even in this case. We also show that the proposed platform can potentially support complex and demanding workloads, which justify its usage in the smartwatch platform we deployed as well as in other smart wearable devices. In fact, the proposed platform provides a highly effective accelerator that could be exploited for other emerging wearable applications, to perform low power classification directly on-board of wearable devices [16][17]. Our claims are *i*) that the availability of more computing power enables extraction of more complex features out of the same simple ultra-low power sensors; and *ii*) that our platform can support a workload orders of magnitude more complex than what can be supported by current off-the-shelf wearables, within a similar power envelope.

The paper is organized as follows: Section II describes related work; Sections III and IV detail the system architecture and classification approach; Section V describes our results.

II. Related work

Due to the need for performance that is typical of many approaches based on machine learning, most research on wearable sensor systems has focused on smartphones, that provide an ideal platform from this point of view as they provide a personal portable, sensor-rich and powerful computing platform [18][19]; they can also be used as a hub for a network of smaller sensors. Using the MEMS sensors embedded in most modern smartphones it is possible to perform tasks such as activity recognition, crowd sensing and fall detection with great effectiveness [6][20], using classification techniques such as decision trees, k-nearest neighbors, support vector machines (SVMs), naïve Bayes and neural networks [21]. For example, Porzi et al. [11] build a wearable system for gesture recognition to help visually impaired using a Sony Xperia Z smartphone and a Sony Smartwatch. They make use of an optimized kernel method (global alignment kernel) for discrete-time warping in SVMs, allowing to map similar gestures when moving at different speeds.

However, a smartphone-based wearable may not be the best choice, due to its limited battery duration and the requirement of wireless connection with the body sensors, non real-time operation (as it depends on the complex operating system running on the phone) and loose coupling with the body (e.g. it is easy to forget the phone anywhere). The main alternative for body sensing is based on low-power microcontrollers [1] that usually run either bare-metal code or a very small real-time operating system such as FreeRTOS. Examples of ultra-low power microcontrollers that are able to work in a power budget of less than 50 mW include the SiliconLabs *EFM32* [22], the Texas Instruments *MSP430* [23] series of MCUs, the *Ambiq Apollo* [24], and the STMicroelectronics STM32-L476 [25]. A typical approach is to employ a heterogeneous set of sensors such as accelerometers, acoustic sensors, gyroscopes and thermometers on the human body to capture characteristic repetitive motions, postures, and sounds of activities [26] that can then be used for context classification. Battery-less and/or harvesting-based systems including several sensors and based on simple microcontrollers have been recently proposed in literature [27][28][29]. These solutions typically enable efficient data collection but onboard microcontrollers are capable of only minimally complex data analytics, needing an external computing platform (smartphone, cloud) if more complex computation is needed, reducing the portability and constraining the usability of the system.

Many wearable systems do not include cameras because it is difficult to extract meaningful data out of them while keeping a very tight power and energy budget. On the other hand, it is well known that cameras are a very effective source of information regarding one's own body [30][9], especially

taking advantage of the preferential ego-vision point of view. To exploit this richness under the tight energy constraints, it is necessary to couple a very efficient imaging sensor with a computing platform that can provide enough throughput to extract significant information out of the frames. Research on ultra-low power cameras focuses on relatively small gray-scale imagers [31][32][33]. These cameras often output analog pixels, needing an external ADC to convert the frames to the digital domain and complicating the classification task due to the amount of noise. This further strengthens the need for a relatively high-performance computing platform to be embedded in the sensor node.

To try and overcome the energy efficiency limitations of current commercial ultra-low power platforms, researchers have to extract as much energy efficiency as possible out of silicon. A well known approach is *near-threshold computing*, that exploits the fact that CMOS technology is most efficient when operated near the voltage threshold, where delay and dynamic power are simultaneously small and therefore total energy per operation is minimal [34]. For example Ickes et al. [35], *SleepWalker* [36] and *Bellevue* [37] show examples of near-threshold ultra-low power microcontrollers, with the latter also exploiting SIMD parallelism to improve performance.

Microcontrollers can also exploit accelerators as ASICs [1][38] to achieve a higher level of performance; however, such approaches are very limited in flexibility, which negatively impacts economy of scale and cost. Instead, a key enabler to achieve high performance with little or no sacrifice to flexibility is *parallel computing*, which is an attractive option for highly parallel workloads such as those of computer vision. Operating multiple cores in parallel allows for the inherent data- and task-parallelism of the algorithm at hand to be exploited, while the energy costs of the platform are partially shared between the cores improving overall efficiency. Traditionally, in the embedded world parallelism has been exploited by means of special-purpose DSPs relying on SIMD or VLIW. Two examples are the Qualcomm Hexagon DSP [39] that accelerates a Snapdragon 800 with VLIW DSPs and is effective for vision and context inference tasks [40], as well as the Neon SIMD extensions that are integrated in many ARM cores [41]. All these platforms, however, are not meant to couple with a low power microcontroller, as they are designed for high end embedded architectures with DRAM, memory management and complex operating systems with power budget in the hundreds of milliwatts at chip level, up to a few watts at system level.

Table I shows an overview of some state-of-the-art activity recognition works. The proposed algorithms target fall detection using the camera sensor as main device, coupled with low power computational resources. In contrast with our work, neither of the two architectures is based on a low-power microcontroller. CITRIC [42] is based on the Intel XScale microarchitecture (with ARMv5 ISA) running at about 600 MHz. It was initially developed as a standalone video processing node. Exynos 5410 Octa [43] is a commercial system-on-chip by Samsung that can be found in several smartphones such as the Samsung Galaxy S4. It is based on an ARM big.LITTLE architecture and contains 4 Cortex-A7 and 4 Cortex-A15 cores (with SIMD extensions) plus a PowerVR SGX544 GPU. Compared to our work, the considered platforms require order of magnitude more power, while targeting a similar class of algorithms in terms of computational requirements.

Algorithm	Architecture	Accuracy	Power
HOG [13]	CITRIC[42]	87%	$\sim 1\,\mathrm{W}$[42]
Optical Flow [13]	CITRIC[42]	85%	$\sim 1\,\mathrm{W}$[42]
Erden et al. [12]	Exynos 5410 [43]	74%	$\sim 3\,\mathrm{W}$[43]

Table I: Order of magnitude of power consumption and average accuracy in fall detection and activity classification for several related works.

More recently, research has been very active on exploitation of intrinsic data and task parallelism with sub-$100\,\mathrm{mW}$ multi-core platforms; by coupling parallel computing with low power techniques such as near-threshold computing, it is possible to maximize the overall energy efficiency of a platform. Fick et al. [7] propose *Centip3de*, a large-scale fabric of clusters of 64 Cortex M3 cores, integrated in a 3D matrix and clocked at a very low frequency of $10\,\mathrm{MHz}$; it can reach a peak performance of 0.64 GOPS. Another similar platform is *DietSODA* [44] that features 128 SIMD lanes working at relatively low frequency ($50\,\mathrm{MHz}$), reaching up to 6.4 GOPS. On the commercial side, NXP has recently proposed an asymmetric dual-core microcontroller, the NXP LPC54100 [45], that couples a low-power Cortex-M0 for sensor control with a more powerful Cortex-M4 that can be seen as an accelerator.

Our work focuses on enabling high-level visual feature extraction in a low power wearable device. To this end, we augment a low power smartwatch platform with a parallel ULP programmable accelerator that was designed according to the two guidelines that were described with regard to the related work: near threshold and parallel computing. Our first objective is to provide a platform that allows for efficient context classification using visual features at a low power and energy budget; moreover, we want to demonstrate how such a platform can enable many future developments in the fields of vision and ego-vision embodied in low power wearable devices.

III. Smartwatch System Architecture

This section describes the system architecture of the proposed smartwatch, whose high-level diagram is shown in Figure 1. The smartwatch is composed of a low power micro-controller coupled with an ultra-low power accelerator and a set of four different sensors: camera, microphone, accelerometer and thermistor. The proposed architecture extends a previous work by Magno et al. [46] that did not include the ultra-low power accelerator.

The main system runs on a $2\,\mathrm{V}$ power supply, powered by a power harvester BQ25570 from Texas Instruments. The power harvester is connected to a lithium-ion polymer rechargeable battery and can harvest from solar cells and thermal electric generators (TEGs). For the camera and for the microphone additional supply voltages are needed; the microphone is

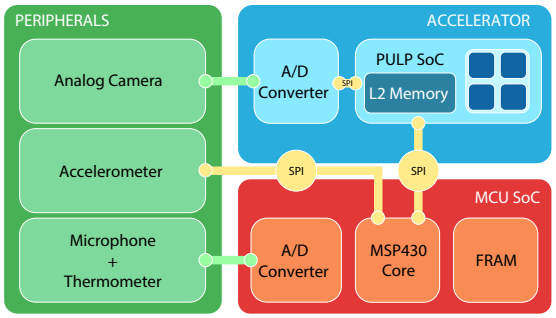


Figure 1: Smartwatch system architecture.

supplied at 1.2 V by a Linear Technologies LTC3406ES5-1.2 buck converter featuring only 1 µA leakage in active mode and the camera with a buck converter TPS62740 (with quiescent current of 460 nA) from Texas Instruments. In idle mode, all sensors can be switched off: camera and microphone are power-gated and controlled by the microcontroller. The accelerometer features a very low-power idle mode that can be set by the microcontroller, and has wake-up by interrupt capability. During idle mode the microcontroller can be put in ultra low power mode or deep sleep, waiting respectively on SPI communication or alternatively on a pin interrupt.

A. *MSP430 core*

The central core of the smartwatch is the 16-bit *MSP430FR5969* microcontroller from Texas Instruments [23]. This microcontroller incorporates 2kB of SRAM and 64kB of non-volatile Ferroelectric RAM (FRAM). The MSP430 is well known for its ultra-low power consumption as it supports several power modes (one active mode and seven low-power modes), enabling fine-grain control of which components of the MCU are active. Current consumption in active mode is of 800 µA at a clock frequency of 8 MHz; this drops to 20 nA in low power mode *LPM4.5*.

B. *PULP accelerator*

In this work, we augment the smartwatch with an accelerator based on the *PULP* platform, a scalable clustered multicore designed to achieve high energy efficiency over a wide range of application workloads [47][8]. In particular, we focus on PULPv3, the third embodiment of the PULP architecture, fabricated in 28 nm FD-SOI ; we emulated this version of PULP with a RTL-equivalent FPGA emulator based on a Xilinx Zynq Z-7045 device. It features a quad-core cluster integrated with 128 kB of L2 SRAM memory and several IO peripherals accessible through a system bus such as two QSPI interfaces (one master and one slave), GPIOs, a bootup ROM and a JTAG interface suitable for testing. The QSPI interfaces can be configured in *single* or *quad* mode depending on the required bandwidth, and they are suitable for interfacing the SoC with a host microcontroller such as the MSP430. In our smartwatch platform, the MSP430 acts as an SPI master with respect to PULP allowing to offload code and data and to control the accelerator. Additionally, two interrupt lines (one per direction) can be used to notify the accelerator or the host (respectively) of a notable event, e.g. to wake up the accelerator or to notify the host of the completion of an accelerated task. The architecture of the PULPv3 SoC is shown in Figure 2).

The PULP cluster is based on 4 OpenRISC-ISA cores with a power-optimized microarchitecture called OR10N [48] and a shared instruction cache (*I$*). The OR10N core is enhanced with respect to the original OpenRISC reference implementation by adding a register-register multiply-accumulate instruction, vectorial instructions for arithmetic on `short` and `char` vectors, two hardware loops and support for unaligned memory access. To avoid the energy overhead of memory coherency, the cores have no data cache and no private L1 memory: they all share a multi-banked tightly coupled data memory (*TCDM*) that acts as a shared scratchpad at L1 [49]. The TCDM is further divided in SRAM and standard-cell memory (*SCM*) banks to allow the cluster to work at very low voltage [50]. A lightweight multi-channel DMA directly connected to the TCDM can be used for fast communication with the L2 memory and external peripherals [51]. The PULP platform is fully programmable using the standard OpenMP programming model [8], which enables relatively easy implementation of parallel algorithms leveraging a low-overhead runtime.

To enable fine grained frequency tuning, a Frequency-Locked Loop [52] and two clock dividers (one for the cluster and one for peripherals) are included in the SoC. All cores use the same clock, but they can be separately clock-gated to reduce dynamic power or boosted with body biasing. A HW synchronizer helps synchronization between the cores and manages sleep states and clock gating in a fast, centralized fashion. This feature is directly integrated in the threading runtime and transparent to the user.

Figure 3 clarifies in a quantitative way why PULP is a highly effective accelerator for highly power constrained microcontroller level systems. The plot shows the power consumption of several low-power MCUs (including the MSP430) and of PULP against their peak throughput in terms of operations per second. The operating points taken into account include all supply voltages from V_{DD} =0.5 V to V_{DD} =1.0 V in 100 mV steps. In the case of the MCUs the operating points are chosen from those reported in their data sheets, while for PULP they are those considered during

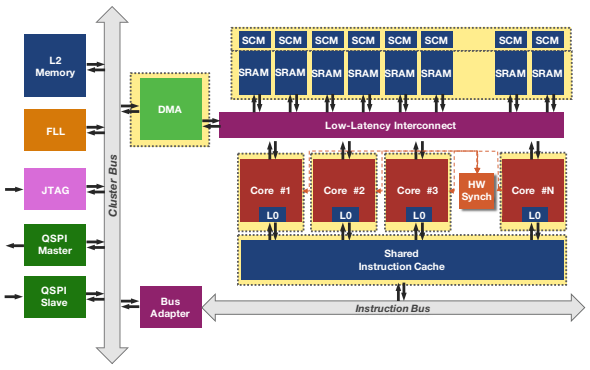


Figure 2: PULP System-on-Chip architecture.

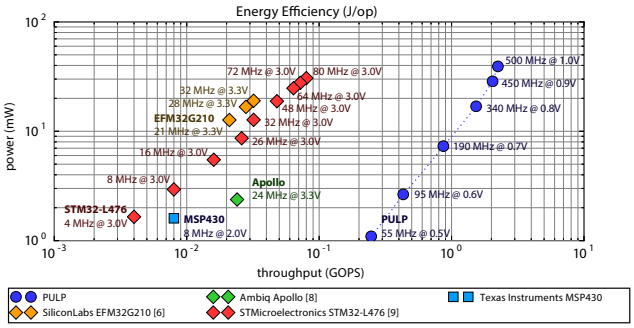


Figure 3: Power consumption and performance of MSP430, PULP and several commercial MCUs.

power analysis (see Section V). Figure 3 takes into account 4 state-of-the-art low-power microcontrollers: Texas Instruments MSP430 [23], SiliconLabs EFM32 [22], Ambiq Apollo [24] and STMicroelectronics STM32-L476 [25]; the latter two feature a relatively powerful ARM Cortex-M4 core. By comparing the MCUs and PULP in several operating points, the plot highlights the trade-off between the two kinds of platforms: on one hand, PULP relinquishes many features as a MCU such as those for interfacing with many different kinds of analog sensors and ultra-low current duty-cycle management; on the other hand, thanks to its architecture optimized for parallel and near-threshold computing and to its deeper integration technology, PULP is vastly more energy efficient than the MCUs. This energy efficiency margin can be used to provide the same performance at a lower power cost, or higher throughput within the same power envelope, and it is essentially the necessary precondition for acceleration [53].

C. Sensors

The smartwatch hosts four different sensors. The first sensor is an ultra-low power analog gray-scale 112×112 *Centeye Stonyman* CMOS camera [33], which has a focal plane size of $2.8\,\mathrm{mm}\times2.8\,\mathrm{mm}$ and a pixel pitch of $25\,\mu\mathrm{m}$ in an active power envelope of $2\,\mathrm{mW}@3.3\,\mathrm{V}$ (with quiescent power as low as $30\,\mathrm{nW}$). The camera can take a new picture every $\sim 50\,\mathrm{ms}$. The brightness values of each pixel is read out row by row while the pixel address is changed by short pulses on the control input pins. As the camera is intended for ultra-low power application, the camera does not do any on-chip preprocessing (e.g. automated exposure adjustment). The camera comes on a pre-soldered PCB containing the image sensor and a lens and is connected to the smartwatch by a socket connector. The camera is plugged directly to the PULP vision accelerator via an *ADS7042* ADC, as shown in Figure 1, while the other sensors are plugged to the MSP430 microcontroller via SPI (accelerometer) and the internal ADC of the MSP430 (microphone, thermometer).

The accelerometer is an ultra-low power *ADXL362* from Analog Devices with high resolution (down to $9.8\,\mathrm{mm/s^2}$). While sensing at $100\,\mathrm{Hz}$, it needs $1.8\,\mu\mathrm{A}$ at a supply voltage of $1.8\,\mathrm{V}$, which are reduced to $10\,\mathrm{nA}$ in standby mode. The accelerometer features a burst mode including a FIFO buffer, that allows to store the acquired sensor data inside the sensor while keeping the MCU asleep. To connect the MCU to the accelerometer, the SPI interface is used with the addition of two status signals that can be used to interrupt or wake-up the microcontroller, e.g. when acceleration exceeds a predefined threshold or the FIFO buffer is full. As a microphone, the smartwatch board includes the low power *INMP801* which was mainly designed for hearing aids and consumes $17\,\mu\mathrm{A}$ at a supply voltage of $1.2\,\mathrm{V}$, with an output voltage in the range of $410\,\mathrm{mV}$-$730\,\mathrm{mV}$. The audio signal is amplified by a *TI LMV951*, connected to the internal ADC of the MSP430, which is set to sample the audio signal at $8\,\mathrm{kHz}$. Finally, the temperature sensor is a Negative Temperature Coefficient Thermistor (NTC) from Epcos/TDK used in a voltage divider configuration and is also connected to the ADC of the MSP430. The temperature sensor is directly supplied by an output pin from the microcontroller such that power is only consumed when temperature is measured and no additional load-switch is needed.

IV. Context Classification

In this section, we describe the techniques that were used to extract features out of the various sensory data and to classify it in one of several contexts. As target platforms, we consider both the non-accelerated smartwatch introduced in Magno et al. [46] and the accelerated version we described in Section III. As a demonstration of a context classification application, we used the features extracted to infer whether the smartwatch user is in one of five "contexts": *morning preparation*, *walking outdoors*, *public transportation*, *in the car* and *in the office*. The full dataset used for training the classifiers comprised $\sim$35000 data points, each including an image acquired from the Stonyman camera and data from the other sensors. The dataset was collected by a total of 15 people wearing a smartwatch prototype for a combined total of 15 hours in different contexts corresponding to the five classes. Acquired images, recorded audio and temperature and acceleration measurements were captured synchronously and kept correlated within the dataset by timestamping them. Data was divided in time frames of $1\,\mathrm{s}$, overlapped by $500\,\mathrm{ms}$. A single camera shot is shared between 5 frames (a $0.4\,\mathrm{Hz}$ rate), audio/accelerometer acquisition is continuous ($8\,\mathrm{kHz}$ and $100\,\mathrm{Hz}$ rates, respectively) and there is a single thermal measurement per frame (at $2\,\mathrm{Hz}$). All data was fed within the various algorithms we describe in Sections IV-A and IV-B with no preliminary preprocessing. Figure 4 shows an example data point for the temperature, accelerometer and camera sensors.

A. Feature Extraction on the MSP430

The first step of context recognition is extracting features out of raw sensor data. To this end, the data is fed into an algorithm that collapses it into a compact feature space by means of a reduction operation; one of the simplest conceivable features is for example the average of all inputs. Most algorithms, such as SVMs and CNNs, use a more complex technique to extract features, by first projecting the input data into an intermediate

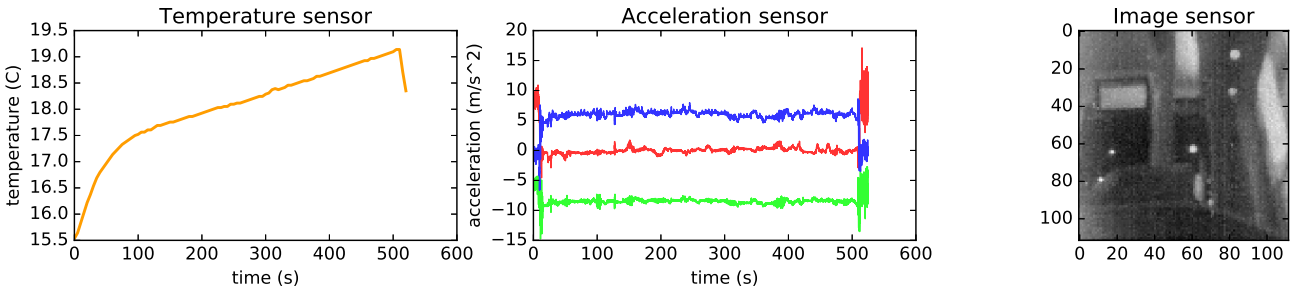


Figure 4: Example of the temperature, accelerometer and camera sensor outputs from the dataset.

high-dimensional space where the selected features are linearly separable and can be more easily extracted. If the features are selected correctly, the final classifier (e.g. the context classifier in our case) can be simpler and more effective; however, in the case of the proposed smartwatch, it is necessary to trade off the necessity to extract high level features against the limited available computing capability.

1) Camera: Vision sensors in a smartwatch can potentially produce a huge amount of useful data on the person wearing it. However, extraction of high-level features is not possible on low power microcontrollers used in wearable devices, as the MSP430, due to the computational burden of complex feature extractors used in the machine vision field. As a consequence, we consider only very simple features to be computed on the MSP430. In the context of this work, we consider three features: pixel *average intensity*, *intensity variance* and *max-min difference*.

2) Accelerometer: The accelerometer is widely used in many applications, being generally recognized as one of the most important sensor providing contextual information; when mounted on a smartwatch, it can be used to distinguish the type of activity that the user is doing (e.g. drinking a coffee, typing, etc), and hence the most probable context he is in. For each of the acceleration directions, we define two main features: *energy*, defined as the cumulative square sum of acceleration over a window of samples; and *acceleration entropy*, defined as

$$H_{\text{accel}} = \sum_{i=0}^{N-1} \left(|\hat{a}_i| \cdot \log_2 (\hat{a}_i) \right) \quad (1)$$

where $\hat{\mathbf{a}}$ is the normalized acceleration.

3) Microphone: The microphone is a powerful sensor to distinguish one context from another, because every environment can differ in its audio characteristics. The first audio feature we considered is the *zero-crossing rate* on frames of the duration of 0.5 seconds, as a first-order approximation of the tone pitch. The other features depend on a frequency domain representation of the audio signal; we used a 1024 point *Fast Fourier Transform (FFT)* both as a feature itself and to compute a set of higher level features: 16 *Mel Frequency Cepstrum* (MFC) coefficients, which represent the human ear perception of a given physical frequency and are obtained by Dirichlet bandpass filtering of the frequency-domain audio signal.

4) Temperature: Temperature helps distinguishing outdoor from indoor environments in a given season. Moreover the corresponding sensor has by far the lowest power consumption, which makes it even more attractive. The only feature of interest we considered is the *average* over a window of samples.

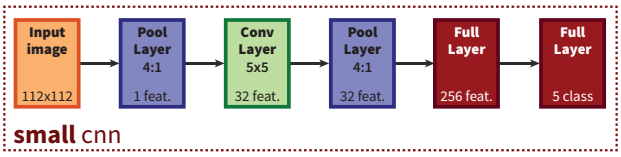


Figure 5: *small* CNN architecture for feature extraction on PULP.

B. Visual Feature Extraction on PULP

The availability of the PULP accelerator makes it possible to implement much more complex feature extractors. In particular, the information coming from the camera is decidedly under-utilized in the MSP430 due to sheer amount of computations that would be necessary to extract complex features from an image. Conversely, PULP is well-suited for acceleration of vision kernels due to the amount of algorithmic parallelism available. In the accelerated smartwatch, we can afford to augment or replace the three features available for the camera (average, variance, max-min difference) with more complex algorithms.

In particular, we focused on a simplified version of a feature that is usually available in higher level computer vision platforms: a Convolutional Neural Network (*CNN*) [54]. Full-fledged CNNs are state-of-the-art in many current visual classification, detection and scene understanding benchmarks using big networks designed to run on relatively high performance platforms such as GPUs [55][56][57]. However, in this case (as is shown in Figure 5) we consider a very small CNN architecture that begins with a strong reduction in the dimensionality of the input (using a 4:1 max-pooling layer) to reduce the computational complexity of the model. Our CNN implementation is based the *CConvNet* library [43], that takes advantage of the OpenMP programming model for better performance on the parallel PULP platform.

C. Sensor fusion and Classification

The sensor fusion and classification stage is based on a Decision Tree (DT), one of the simplest and most widely

applied supervised classification techniques [58]. We selected this technique in particular because of the need of an algorithm with low computational complexity and high energy efficiency in inference, constraints that made the DT a suitable choice for our specific domain. We use the Decision Tree as the final classification stage, feeding it with all features described in Sections IV-A and IV-B. Inference works by exploring the tree, starting from the root node until one of the leaf nodes is reached which point to the most probable activity class. During the tree traversal for classification each node compares the value of its associated feature to a pre-learned threshold to decide on which branch to take next.

The specific algorithm we used to create the tree is based on the continuous C4.5 algorithm [59], resulting in a single tree that takes in account all the features evaluated by the MSP430 and by PULP. The C4.5 algorithms creates a decision tree which is iteratively composed of nodes with four attributes: feature f, threshold T and two children nodes. When used for inference, the C4.5 algorithm starts at the root evaluating the value of its feature f_{root}; then, depending on whether the computed f_{root} is smaller or bigger than the threshold T_{root}, it continues with the left or right child node. This procedure is continued until a leaf node is reached; this node is tagged with the most probable context class. For the supervised learning C4.5 uses a divide and conquer technique. The C4.5 algorithm tries to split the dataset into two subsets with as much information content as possible, i.e. with the activity classes as uniform as possible in each subset; the measure of this uniformity is entropy in the sense of information theory. We defer to Quinlan et al. [59] for the detailed learning algorithm explanation.

A possible drawback of the C4.5 algorithm is represented by overfitting, which can derive from the usage of continuous-valued features and from the limited amount of training data available in the dataset. To limit this phenomenon, we used a top-down pruning approach [60].We used leave-one-out cross-validation [61] for evaluation. Thus, for each collected sequence of activity a decision tree was trained based on the full set excluding the test sequence; the results we present in Section V are averaged over all test sequences.

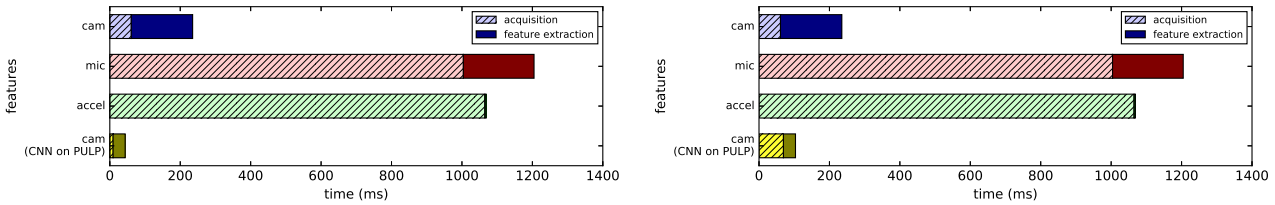
V. Results

In this section we evaluate the accelerated smartwatch platform in terms of power and execution time, as well as in the accuracy of the context classification task. As a term of comparison, we use the non-accelerated smartwatch proposed in Magno et al. [46]. MSP430 code was compiled using the `ti-cgt-msp430` 4.4.6 toolchain, while for PULP we used a custom OR10N toolchain, based on GCC 5.2. We estimated power consumption for PULP using backannotated switching activities from three input vectors in power analysis: *idle*, *matmul* (which simulates a case where the cores are all running, with a low pressure on the shared memory) and *dma* (which simulates a case where the DMA is running, with high pressure on memories). Then, we run our tests on an FPGA-based emulation platform for PULP [53], collecting active and idle cycles for cores, DMAs and interconnects. We model leakage power, dynamic power density and maximum clock frequency at each operating point after the post-layout backannotated timing and power analysis results for the latest PULP chip. For this purpose, we considered the $V_{\mathrm{DD}} = 0.5\,\mathrm{V}$ operating point, that shows the best energy efficiency according to Figure 3. In this operating point, f_{clk} is $50\,\mathrm{MHz}$. The power consumption for the MSP430 and the peripherals were measured during idle and active mode where the microcontroller was supplied by $2\,\mathrm{V}$ and was operating at $8\,\mathrm{MHz}$.

A. Context classification

To compare the non-accelerated platform of Magno et al. with our proposed PULP-accelerated platform, we considered a set of combinations of several feature extractors, fused inside the decision tree as explained in Section IV-C. In particular, we consider the following feature extractors: `temp`, `cam`, `mic(no fft)`, `mic`, `accel` and their combinations indicate tests using the features described in Section IV-A, which work without using the accelerator in the same way as in Magno et al. [46]. `mic(no fft)` does not include features based on the frequency domain representation of the audio signal, while `mic` includes all audio features. `all(no fft)` and `all` indicate that all the features described in Section IV-A (`temp+cam+mic+accel`) are used (without or with FFT-based features, respectively); in the case of the non-accelerated platform of Magno et al. [46], all of them are executed on the MSP430, whereas in the accelerated platform we execute the extraction of features from the camera on PULP and that of the other features on the MSP430. `cnn` is a test running on the accelerated platform where the classifier is the small CNN described in Section IV-B; in this case the Decision Tree is not used. `all+cnn`, finally, considers the case in which we use the accelerated smartwatch with all non-visual features of Section IV-A extracted on the MSP430, while we also integrate the output of the small CNN of Section IV-B into the Decision Tree.

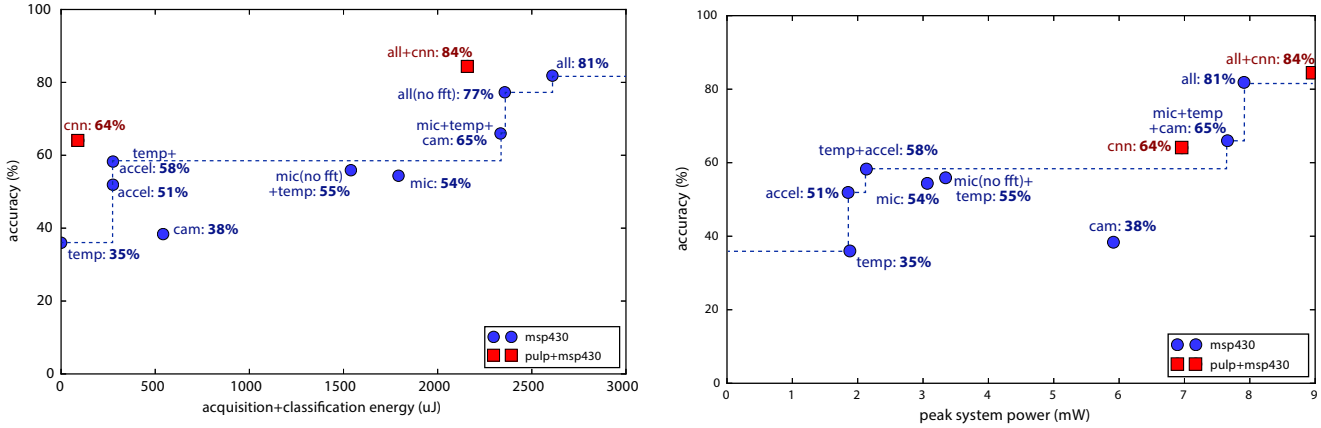
Figures 6a and 6b focus on a preliminary analysis of our baseline, i.e. the non-accelerated platform of Magno et al. [46]. We show time and energy costs of each sensor divided in *acquisition* and *feature extraction* (i.e. computation); the thermistor is orders of magnitude less expensive and is thus not shown. The data shown does not consider the possibility of overlapping sensor acquisition with computation, which would further reduce the overall time. The accelerometer and the microphone need a long time to acquire data (on the order of $1\,\mathrm{s}$), while in the non-accelerated platform the camera is more than $20\times$ faster, taking only $61\,\mu\mathrm{s}$ to acquire data. Similar time/energy are spent in the non-accelerated platform to extract audio and camera features, but while for the former it is possible to extract relatively complex frequency-domain features, for the latter the same energy is spent to extract very simple average-based features. The Figures also report energy/time in the proposed accelerated platform when using the simple CNN of Section IV-B; the external ADC connected to PULP is also more efficient than the internal MSP430 ADC, providing a significant efficiency improvement to the platform. Overall, feature classification energy is reduced by using the PULP accelerator even if the feature extractor is much more complex, as more thoroughly exposed in the following.



(a) Acquisition and feature extraction time per classification.

(b) Acquisition and feature extraction energy per classification.

Figure 6: Time and energy to acquire and elaborate features in the non-accelerated Magno et al. [46] platform and in the proposed platform when using the *small* CNN for camera feature extraction.



(a) Context recognition accuracy vs energy spent per acquisition+classification.

(b) Context recognition accuracy vs peak system power.

Figure 7: Trade-off between context classification accuracy and energy/power, with annotated Pareto boundary for the non-accelerated platform (dashed line).

Figure 7a plots accuracy versus energy per classification for both the platforms being compared. The blue dots in the plot refer to the non-accelerated case of Magno et al. [46], where all computation is performed by the MSP430, while the red ones refer to the PULP-accelerated one. Each dot is tagged with the set active sensors and with the total classification accuracy obtained, and the dashed line highlights the Pareto-dominant points for the non-accelerated platform of Magno et al. [46] in the accuracy-energy tradeoff. As could be expected, a clear tradeoff between accuracy and energy is shown here; it is necessary to spend more energy to obtain a better result in terms of accuracy. It is interesting to observe that of the four points where the camera is used in the non-accelerated platform, two (`mic+cam+temp`, `all`) are Pareto-dominant, clearly indicating that even with the very simple features that can be run on the MSP430 the camera achieves a good level of separation over the five classes considered (*morning preparation*, *walking outdoors, public transportation, in the car, in the office)*; in particular, the fact that the results exceed those obtained with the accelerometer alone confirms that sensor data from the camera can be significant for the context recognition task. The two PULP-accelerated points are both abundantly Pareto-dominant in terms of accuracy per Joule, yielding up to 84% accuracy when using all features and the CNN (`all+cnn` case) while at the same time saving more than $400\,\mu J$ per classification with respect to the best non-accelerated point (`all`). The pure `cnn` case achieves a 64% accuracy comparable to that available when using the audio features in the non-accelerated platform, but at an energy budget per classification that is $25\times$ lower ($\sim 91\,\mu J$).

The two `all` and `all+cnn` points are relatively close in terms of accuracy; adding the CNN we are able to get an additional 3% of average accuracy on the five classes. Although the difference in terms of average accuracy is small, a closer look at the confusion matrices shows that the `all+cnn` case is actually a significant improvement over the `all` one. Figure 8 shows that in the `all` case there are two sources of inaccuracy: confusion between *in the car* and *public transportation*, and confusion between *walking outdoors* and *in the office*. As a consequence, only the accuracies of *morning preparation* and *walking outdoors* are above 90%. The `all+cnn` eliminates the second of these two inaccuracies, bringing the precision of *in the office* above 90%. The confusion between *in the car* and *public transportation* stays also in the `all+cnn` case; however, in our opinion this can be justified by the objective similarity of the two situations (sitting in a bus versus sitting in a car).

Figure 7b expands our analysis with the tradeoff between accuracy and peak power, an important metric for wearable systems as their small batteries are typically limited not only in terms of total energy capacity but also in sustainable power output. Accelerometer and thermistor contribute relatively little to the total system power consumption; the main dominant

costs are therefore the compute units (MSP430 and PULP), the camera and the microphone. The first interesting point to raise is that even when all sensors and compute units are kept on, total system power peaks at ≈9 mW, and that the addition of the PULP accelerator increases this peak power by less than 15% with respect to the peak power consumption of the Magno et al. [46] platform. Moreover, by comparing Figures 7a and 7b, it is easy to observe that even if the peak power consumption in the accelerated platform may be slightly higher, the overall energy consumption (and thus average power) is considerably lower, which means that if the platform is able to provide ∼10 mW of peak power, the accelerated platform is convenient in terms of both energy and average power.

B. Visual feature extraction exploration

The simple visual feature extraction technique that we have described for the use case of Section V-A demonstrates that it is possible to use relatively complex ego-vision feature extraction for the purpose of context classification on a low-power, low-energy consumption platform. However, the example we have shown is far from saturating the capabilities of the PULP accelerator, that could be also used for more complex functionality. In this section, we showcase how the availability of PULP enables the implementation of much more complex vision algorithms to implement more advanced ego-vision tasks while keeping the power envelope within 10 mW to 100 mW. To do this, we expand the set of visual features we consider with two more tests that are more directly inspired to the state-of-the-art in computer vision.

The first additional test is a bigger Convolutional Neural Network, whose architecture is shown in Figure 9. Removing the initial max-pooling stage and adding additional layers, the computational complexity of this CNN is orders of magnitude bigger than that of the one described in Section IV-B, though it is still simpler than most models targeted at high performance platforms, such as AlexNet [55] and GoogLeNet [56].

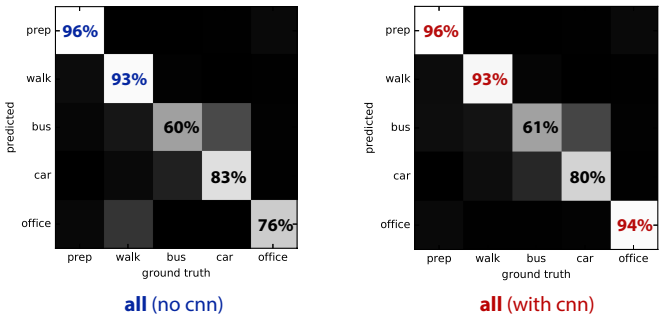


Figure 8: Confusion matrices for `all` and `all+cnn` tests.

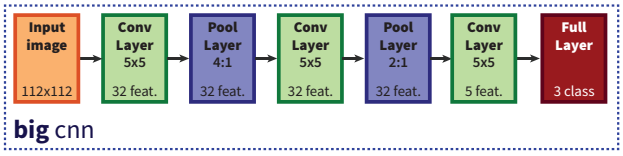


Figure 9: *big* CNN architecture for feature extraction on PULP.

As a second additional test, we implemented an HOG+SVM pipeline. Support Vector Machines trained on Histogram of Oriented Gradient (HOG) features represent a *de facto* standard across many visual perception tasks [62][63][64][65]. For our evaluation, we run on PULP the same version of the HOG algorithm originally proposed in [66]; using a 112x112 pixels image we obtain 784 features, each one evaluated with respect to 9 directions, which also correspond to the number of bins in each histogram. The final descriptor, made up of 7056 elements, is then used as an input to the SVM, where we consider a Gaussian kernel with 256 support vectors for classification. HOG was implemented by extending code from the VLFeat library, presented in Vedaldi et al. [67], parallelized by using the OpenMP programming model and optimized to work on image strides, thus enabling overlap of data transfer and kernel execution.

Figure 10 highlights how the difference in performance/Watt between the MSP430 and PULP (first shown in Figure 3) can be exploited to implement orders-of-magnitude more complex functionality in terms of feature extraction. We show a comparison in terms of energy in logarithmic scale; execution time scales in a similar way; we also show the same tests on a STM32L476 and Ambiq Apollo for enhanced clarity. We consider to operate the MSP430 at 8 MHz, the Apollo at 24 MHz and the STM32 at 80 MHz. For PULP we chose the 0.5 V operating point that was also used in the previous Section, corresponding to a 50 MHz operating frequency. For the microcontrollers, power considers only computation, discarding data acquisition. In our platform, energy consumption is shown split in four contributions: PULP computation, MSP430 code offload (via SPI), data transfer from the ADS7420 ADC, and MSP430 sleep time during the execution on PULP (in LPM4.5 mode while retaining register data). Note that it in the typical use case, the code offload from MSP430 is performed once and amortized over all iterations, as PULP will repeatedly execute the same function. We consider feature extraction of all visual features defined in this section and in Section IV-B: the original features for the non-accelerated platform (`mean`, `variance` and `max-min` difference); the two CNNs (`cnn(small)` and `cnn(big)`); and HOG (divided in the histogram extraction, `hog(hist)`, and the SVM, `hog(svm)`).

Figure 10 clarifies how more complex vision pipelines are within reach of the accelerated platform: the entire `hog` pipeline on PULP takes only 69% more time and 4% more energy than the `variance` test on the MSP430, while being more than 300× more complex in terms of elementary RISC operations. For very small kernels, accelerated execution is less convenient as they are fully dominated by data transfer from the ADC. To fully appreciate the advantage of local on-sensor computation, it is also interesting to compare these results with state-of-the-art techniques for wireless data transmission. The most efficient transmission techniques have been proposed in the context of low-bitrate, low-range biomedical devices [68][69]. The transmitter proposed by Ba et al. [68] can work at up to 4.5 Mb/s with an energy consumption of 0.5 nJ per bit (or even less at 11 kb/s) - which would mean 75 µJ to transfer the 112×112 12-bit image produced by our camera (in the same

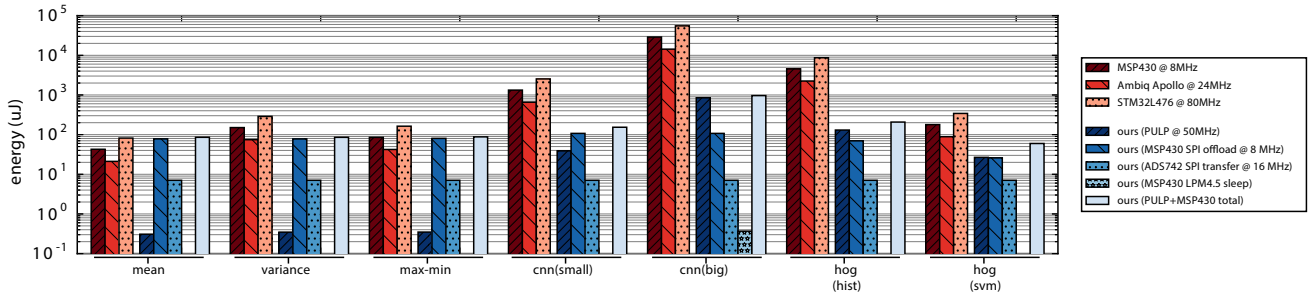


Figure 10: Energy comparison between PULP and several microcontroller platforms.

range of our results). However, these techniques are limited to extremely low range communication, requiring a secondary battery-powered device (e.g. a smartphone) to route data to the cloud using a long range technique. In contrast, long range transmission technologies for the IoT, such as LoRaWAN, are at least $100\times$ slower and consume up to $10\times$ more power [70], resulting in an overall energy consumption on the order of $10\,\mathrm{mJ}$ or more. This fully justifies our local computation approach, which requires less then $1\,\mathrm{mJ}$ even for the most complex benchmark (`cnn(big)`).

C. Battery lifetime estimation

As mentioned in Section III, the system is supplied with two harvester sources (TEGs and solar cells). On average, these sources are able to provide $\sim 41\,\mu\mathrm{W}$, while the system power in deep sleep mode (with the MSP430 in LPM4 mode and PULP and peripherals power-gated) is $38\,\mu\mathrm{W}$. Assuming that the platform mounts a small lithium-ion polymer $4\,\mathrm{V}\ 150\,\mathrm{mA\,h}$ battery, in Table II we estimate the expected lifetime, knowing the energy per acquisition from Section V-A ($2.6\,\mathrm{mJ}$ for `all`, $2.2\,\mathrm{mJ}$ for in `all+cnn`).

	Harvesting	`all`	`all+cnn`
idle (LPM4.5)	No	661d	661d
always on	No	9d	11d
every minute	No	307d	333d
once a day	No	660d	660d
always on	Yes	9d	11d
every minute	Yes	617d	732d
every 14m	Yes	∞	∞

Table II: Lifetime evaluation

Apart from the benefit in accuracy, the accelerated platform is also beneficial in terms of battery lifetime. This benefit steadily grows as we increase the interval between consecutive acquisitions, up to complete autonomy (with harvesting) if the interval is $14\,\mathrm{min}$ or more.

VI. Conclusions

This work demonstrates the importance of vision in context recognition for wearable applications and how it is possible to extract meaningful features out of an ego-vision ULP camera even when working in a very tight power envelope. Using the PULP programmable accelerator, we enable the implementation of vision algorithms of significant level of complexity, while keeping the overall system power budget below $10\,\mathrm{mW}$ at peak. Our results have shown that, leveraging a speedup as high as $500\times$ on the computation of visual features, the heterogeneous platform we propose can achieve the same accuracy than our baseline [46] with a $25\times$ reduction in energy cost; or alternatively a significant accuracy improvement, with 84% average correctness at $2.2\,\mathrm{mJ}$ per classification. Such a platform could be deployed directly on the human body (on wearables such as watches, glasses, necklaces) and provide a small, unintrusive device and with no need of mediation through a smartphone, benefiting applications such as context detection and advanced human interfaces through ego-vision techniques. Moreover, it could constitute the "personal hub" of a complex multi-device system where body-coupled sensors cooperate with resident environmental IoT devices for enhanced context understanding.

References

[1] H. Ghasemzadeh and R. Jafari, "Ultra Low-power Signal Processing in Wearable Monitoring Systems: A Tiered Screening Architecture with Optimal Bit Resolution," *ACM Trans. Embed. Comput. Syst.*, vol. 13, pp. 9:1–9:23, Sept. 2013.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, First 2014.

[3] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *International Workshop on Wearable and Implantable Body Sensor Networks, 2006. BSN 2006*, pp. 4 pp.–116, Apr. 2006.

[4] S. Sharma, J. Agrawal, S. Agarwal, and S. Sharma, "Machine learning techniques for data mining: A survey," in *2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–6, Dec. 2013.

[5] C. Doukas and I. Maglogiannis, "Managing Wearable Sensor Data through Cloud Computing," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 440–445, Nov. 2011.

[6] A. Anjum and M. Ilyas, "Activity recognition using smartphone sensors," in *2013 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 914–919, Jan. 2013.

[7] D. Fick, R. G. Dreslinski, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu, M. Wieckowski, G. Chen, T. Mudge, D. Blaauw, and D. Sylvester, "Centip3De: A Cluster-Based NTC Architecture With 64 ARM Cortex-M3 Cores in 3D Stacked 130 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 104–117, Jan. 2013.

[8] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini, "PULP: A Ultra-Low Power Parallel Accelerator for Energy-Efficient and Flexible Embedded Vision," *Journal of Signal Processing Systems, to appear.*

[9] G. Serra, M. Camurri, L. Baraldi, M. Benedetti, and R. Cucchiara, "Hand Segmentation for Gesture Recognition in EGO-vision," in *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*, IMMPD '13, (New York, NY, USA), pp. 31–36, ACM, 2013.

[10] Z. Lv, L. Feng, H. Li, and S. Feng, "Hand-free Motion Interaction on Google Glass," in *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, SA '14, (New York, NY, USA), pp. 21:1–21:1, ACM, 2014.

[11] L. Porzi, S. Messelodi, C. M. Modena, and E. Ricci, "A Smart Watch-based Gesture Recognition System for Assisting People with Visual Impairments," in *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*, IMMPD '13, (New York, NY, USA), pp. 19–24, ACM, 2013.

[12] F. Erden, S. Velipasalar, A. Z. Alkar, and A. E. Cetin, "Sensors in assisted living: A survey of signal and image processing methods," *IEEE Signal Processing Magazine*, vol. 33, pp. 36–44, March 2016.

[13] K. Ozcan, A. K. Mahabalagiri, M. Casares, and S. Velipasalar, "Automatic fall detection and activity classification by a wearable embedded smart camera," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 3, no. 2, pp. 125–136, 2013.

[14] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, pp. 579–590, May 2013.

[15] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, Sept. 2013.

[16] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, 2015.

[17] K. Bastani, S. Kim, Z. . Kong, M. A. Nussbaum, and W. Huang, "Online Classification and Sensor Selection Optimization With Applications to Human Material Handling Tasks Using Wearable Sensing Technologies," *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 485–497, Aug. 2016.

[18] M. Azizyan and R. R. Choudhury, "SurroundSense: Mobile Phone Localization Using Ambient Sound and Light," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, pp. 69–72, June 2009.

[19] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility Prediction-based Smartphone Energy Optimization for Everyday Location Monitoring," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, (New York, NY, USA), pp. 82–95, ACM, 2011.

[20] K. E. Seong, K. C. Lee, and S. J. Kang, "Self M2M based wearable watch platform for collecting personal activity in real-time," in *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pp. 286–290, Jan. 2014.

[21] E. Gokgoz and A. Subasi, "Comparison of decision tree algorithms for EMG signal classification using DWT," *Biomedical Signal Processing and Control*, vol. 18, pp. 138–144, Apr. 2015.

[22] "SiliconLabs EFM32G210 Datasheet."

[23] "MSP430FR59xx Mixed-Signal Microcontrollers (Rev. E)."

[24] "Ambiq Apollo Data Brief."

[25] "STMicroelectronics STM32L476xx Datasheet."

[26] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome, "Object-Based Activity Recognition with Heterogeneous Sensors on Wrist," in *Pervasive Computing* (P. Floréen, A. Krüger, and M. Spasojevic, eds.), no. 6030 in Lecture Notes in Computer Science, pp. 246–264, Springer Berlin Heidelberg, May 2010.

[27] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "Wispcam: A battery-free rfid camera," in *2015 IEEE International Conference on RFID (RFID)*, pp. 166–173, IEEE, 2015.

[28] L. Spadaro, M. Magno, and L. Benini, "Kinetisee: A perpetual wearable camera acquisition system with a kinetic harvester: Poster abstract," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, p. 68, IEEE Press, 2016.

[29] A. Dionisi, D. Marioli, E. Sardini, and M. Serpelloni, "Autonomous Wearable System for Vital Signs Measurement With Energy-Harvesting Module," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1423–1434, 2016.

[30] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara, "Gesture Recognition in Ego-centric Videos Using Dense Trajectories and Hand Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 702–707, June 2014.

[31] A. Kerhet, M. Magno, F. Leonardi, A. Boni, and L. Benini, "A low-power wireless video sensor node for distributed object detection," *Journal of Real-Time Image Processing*, vol. 2, pp. 331–342, Oct. 2007.

[32] D. Brunelli, A. Tovazzi, M. Gottardi, M. Benetti, R. Passerone, and P. Abshire, "Energy Autonomous Low Power Vision System," in *Applications in Electronics Pervading Industry, Environment and Society* (A. D. Gloria, ed.), no. 289 in Lecture Notes in Electrical Engineering, pp. 39–50, Springer International Publishing, 2014.

[33] "Centeye Stonyman / Haskbill silicon documentation," 2013.

[34] R. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, pp. 253–266, Feb. 2010.

[35] N. Ickes, Y. Sinangil, F. Pappalardo, E. Guidetti, and A. P. Chandrakasan, "A 10 pJ/cycle ultra-low-voltage 32-bit microprocessor system-on-chip," in *2011 Proceedings of the ESSCIRC (ESSCIRC)*, pp. 159–162, IEEE, Sept. 2011.

[36] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "SleepWalker: A 25-MHz 0.4-V Sub-mm2 7-uW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 20–32, Jan. 2013.

[37] F. Botman, J. D. Vos, S. Bernard, F. Stas, J.-D. Legat, and D. Bol, "Bellevue : a 50MHz Variable-Width SIMD 32bit Microcontroller at 0 . 37V for Processing-Intensive Wireless Sensor Nodes," in *Proceedings of 2014 IEEE Symposium on Circuits and Systems*, pp. 1207–1210, 2014.

[38] T. Fujita, T. Tanaka, K. Sonoda, K. Kanda, and K. Maenaka, "Ultra Low Power ASIC for R-R Interval Extraction on Wearable Health Monitoring System," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3780–3783, Oct. 2013.

[39] L. Codrescu, W. Anderson, S. Venkumanhanti, M. Zeng, E. Plondke, C. Koob, A. Ingle, C. Tabony, and R. Maule, "Hexagon DSP: An Architecture Optimized for Mobile Multimedia and Communications," *IEEE Micro*, vol. 34, pp. 34–43, Mar. 2014.

[40] C. Shen, S. Chakraborty, K. R. Raghavan, H. Choi, and M. B. Srivastava, "Exploiting Processor Heterogeneity for Energy Efficient Context Inference on Mobile Phones," in *Proceedings of the Workshop on Power-Aware Computing and Systems*, HotPower '13, (New York, NY, USA), pp. 9:1–9:5, ACM, 2013.

[41] S. K. Teoh, V. V. Yap, C. S. Soh, and P. Sebastian, "Implementation and optimization of human tracking system using ARM embedded platform," in *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS)*, vol. 1, pp. 353–356, June 2012.

[42] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, *et al.*, "Citric: A low-bandwidth wireless camera network platform," in *Distributed smart cameras, 2008. ICDSC 2008. Second ACM/IEEE international conference on*, pp. 1–10, IEEE, 2008.

[43] F. Conti, A. Pullini, and L. Benini, "Brain-inspired Classroom Occupancy Monitoring on a Low-Power Mobile Platform," in *CVPR 2014 Workshops*, 2014.

[44] S. Seo, R. G. Dreslinski, M. Woh, C. Chakrabarti, S. Mahlke, and T. Mudge, "Diet SODA: A Power-Efficient Processor for Digital Cameras," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design - ISLPED '10*, (New York, New York, USA), p. 79, ACM Press, 2010.

[45] "NXP LPC54100 Datasheet."

[46] M. Magno, D. Porcarelli, D. Brunelli, and L. Benini, "InfiniTime: A multi-sensor energy neutral wearable bracelet," in *Green Computing Conference (IGCC), 2014 International*, pp. 1–8, Nov. 2014.

[47] D. Rossi, A. Pullini, M. Gautschi, I. Loi, F. K. Gurkaynak, P. Flatresse, and L. Benini, "A -1.8V to 0.9V Body Bias, 60 GOPS/W 4-Core Cluster in Low-Power 28nm UTBB FD-SOI Technology," in *Proceedings of the 2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (to appear)*.

[48] M. Gautschi, M. Scandale, A. Traber, A. Pullini, A. Di Federico, M. Beretta, G. Agosta, and L. Benini, "Tailoring Instruction-Set Extensions for an Ultra-Low Power Tightly-Coupled Cluster of OpenRISC Cores," in *Proceedings of VLSI-SOC 2015*, 2015.

[49] A. Rahimi, I. Loi, M. R. Kakoee, and L. Benini, "A Fully-Synthesizable Single-Cycle Interconnection Network for Shared-L1 Processor Clusters," in *2011 Design, Automation & Test in Europe*, pp. 1–6, IEEE, Mar. 2011.

[50] A. Teman, D. Rossi, P. Meinerzhagen, L. Benini, and A. Burg, "Controlled placement of standard cell memory arrays for high density and low power in 28nm FD-SOI," in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pp. 81–86, Jan. 2015.

[51] D. Rossi, I. Loi, G. Haugou, and L. Benini, "Ultra-Low-Latency Lightweight DMA for Tightly Coupled Multi-Core Clusters," in *Proceedings of the 11th ACM Conference on Computing Frontiers - CF '14*, (New York, New York, USA), pp. 1–10, ACM Press, 2014.

[52] I. Miro-Panades, E. Beigné, Y. Thonnart, L. Alacoque, P. Vivet, S. Lesecq, D. Puschini, A. Molnos, F. Thabet, B. Tain, K. B. Chehida, S. Engels, R. Wilson, and D. Fuin, "A Fine-Grain Variation-Aware Dynamic Vdd-Hopping AVFS Architecture on a 32 nm GALS MPSoC," *IEEE Journal of Solid-State Circuits*, vol. 49, pp. 1475–1486, July 2014.

[53] F. Conti, D. Palossi, A. Marongiu, D. Rossi, and L. Benini, "Enabling the Heterogeneous Accelerator Model on Ultra-Low Power Microcontroller Platforms," in *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition*, DATE '16, (San Jose, CA, USA), EDA Consortium, 2016.

[54] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *arXiv:1409.4842 [cs]*, Sept. 2014.

[57] R. Girshick, J. Donahue, T. Darrell, J. Malik, and U. C. Berkeley, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[58] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, pp. 1–37, Dec. 2007.

[59] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Elsevier, June 2014.

[60] C. Drummond, R. C. Holte, and others, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11, Citeseer, 2003.

[61] P. Refaeilzadeh, L. Tang, and H. Liu, *Encyclopedia of Database Systems*, ch. Cross-Validation, pp. 532–538. Boston, MA: Springer US, 2009.

[62] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[63] H. Ameur, A. Helali, M. Nasri, H. Maaref, and A. Youssef, "Improved feature extraction method based on Histogram of Oriented Gradients for pedestrian detection," in *2014 Global Summit on Computer Information Technology (GSCIT)*, pp. 1–5, June 2014.

[64] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using Histograms of Oriented Gradients," *Pattern Recognition Letters*, vol. 32, pp. 1598–1603, Sept. 2011.

[65] J. R. R. Uijlings, I. C. Duta, N. Rostamzadeh, and N. Sebe, "Realtime Video Classification Using Dense HOF/HOG," in *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, (New York, NY, USA), pp. 145:145–145:152, ACM, 2014.

[66] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 1, pp. 886–893 vol. 1, June 2005.

[67] A. Vedaldi and B. Fulkerson, "VLFeat: An Open and Portable Library of Computer Vision Algorithms," in *Proceedings of the International Conference on Multimedia*, MM '10, (New York, NY, USA), pp. 1469–1472, ACM, 2010.

[68] A. Ba, M. Vidojkovic, K. Kanda, N. F. Kiyani, M. Lont, X. Huang, X. Wang, C. Zhou, Y. H. Liu, M. Ding, B. Büsze, S. Masui, M. Hamaminato, H. Sato, K. Philips, and H. de Groot, "A 0.33 nJ/bit IEEE802.15.6/Proprietary MICS/ISM Wireless Transceiver With Scalable Data Rate for Medical Implantable Applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, pp. 920–929, May 2015.

[69] M. Rahman, M. Elbadry, and R. Harjani, "An IEEE 802.15.6 Standard Compliant 2.5 nJ/Bit Multiband WBAN Transmitter Using Phase Multiplexing and Injection Locking," *IEEE Journal of Solid-State Circuits*, vol. 50, pp. 1126–1136, May 2015.

[70] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," in *European Wireless 2016; 22th European Wireless Conference*, pp. 1–6, May 2016.

Francesco Conti (M'14) received M.Sc. and Ph.D. degrees in Electronic Engineering from the University of Bologna, Italy, in 2012 and 2016 respectively. He has been an intern at STMicroelectronics in Grenoble, France in 2012 and a visiting researcher at Stanford University, USA and in ETH Zurich, Switzerland in 2015. He currently holds a position as a postdoctoral researcher at the Integrated Systems Laboratory of ETH Zurich and as research assistant at the University of Bologna. His research interests focus on energy-efficient multicores, with an emphasis on architectural heterogeneity and low-energy paradigms for hardware acceleration, such as brain-inspired computing. He has published more than 10 papers in international conferences and journals, and holds a Best Paper Award from IEEE ASAP'14.

Daniele Palossi is a Ph.D. student at the Dept. of Information Technology and Electrical Engineering at the Swiss Federal Institute of Technology in Zurich (ETH Zurich). He received his B.S. and M.S. in Computer Science Engineering from the University of Bologna, Italy. In 2012 he spent 6 months as a research intern at ST Microelectronics, Agrate Brianza, Milano, working on 3D computer vision algorithms for the STM STHORM project. In 2013 he won a one-year research grant at the University of Bologna, with a focus on design methodologies for high-performance embedded systems. He is currently working on energy-efficient algorithms for autonomous vehicles and advanced driver assistance systems.

Renzo Andri is currently a Ph.D. student in the Integrated Systems Laboratory of the Dept. of Information Technology and Electrical Engineering in ETH Zurich, Zurich, Switzerland, after having received his M.Sc. degree in Electronic Engineering from the same university in 2015. His main research interests involve the design of hardware accelerators for CNNs and the applications of machine learning to energy-efficient digital systems.

Michele Magno (SM'13) received his Masters and Ph.D. degrees in electronic engineering from the University of Bologna, Italy, in 2004 and 2010 respectively. Currently he is a Postdoctoral researcher at ETH Zurich, Switzerland, and research fellow at the University of Bologna, Italy. The most important themes of his research are on wireless sensor networks, low power hardware/software co-design, wearable and embedded video systems. He has collaborated with several universities and research centers, such as the University College Cork and Tyndall Institute, Ireland, Imperial College London, UK, University of British Columbia, Canada among others. He has published more than 60 papers in international journals and conferences.

Luca Benini (F'07) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1997. He is a Full Professor with the University of Bologna, Bologna, Italy, and the Chair of Digital Circuits and Systems with ETH Zurich, Zurich, Switzerland. He has served as the Chief Architect for the Platform2012/STHORM project with STMicroelectronics, Grenoble, France, from 2009 to 2013. He has published over 700 papers in peer-reviewed international journals and conferences, four books, and several book chapters. His current research interests include energy-efficient system design and multicore SoC design. He is also active in the area of energy-efficient smart sensors and sensor networks for biomedical and ambient intelligence applications.