CIRCUIT AND ALGORITHM DESIGN TO ENABLE EDGE INTELLIGENCE

A Dissertation
Presented to
The Academic Faculty

By

Ningyuan Cao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2020

CIRCUIT AND ALGORITHM DESIGN TO ENABLE EDGE INTELLIGENCE

Approved by:

Dr. Arijit Raychowdhury, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Sung-kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Madhavan Swaminathan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Muhannad S Bakir
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Shreyas Sen
School of Electrical and COmputer Engineering
Purdue University

Date Approved: June 27, 2020

Unless he is indifferent to fame and fortune, he cannot have aspirations; unless he stays calm and quiet, he cannot reach afar.

Zhuge Liang, a Chinese politician during Three Kingdom period

For the young man may who stays curious and passionate along the journey.

ACKNOWLEDGEMENTS

I would like to thank the members of my thesis committee for their help in preparation of this work – Dr. Sung-kyu Lim, Dr. Madhavan Swaminathan, Dr. Shreyas Sen and Dr. Muhannad S Bakir. And beyond this work, the committee members have also been so kind to support me in my academic faculty job search and award applications. Without their help, I can not go this far both as a Ph.D. candidate as well as a young man pursuing academic opportunities.

I would like to also thank my colleagues from ICSRL, especially early members: Saad, Samantak, Ahbinav, Anvesha, Insik and Muya. The discussions with them from chip tape-out to research selections are important to the successs of my research endeavors. And the enjoyable morning coffee hours with my colleagues have become the most precious memory.

Special thanks to my advisor Dr. Arijit Raychowdhury. As an understanding and knowledgeable advisor, he has never given me any hard times but instead uses his insights to motivate me and help me reach my goals. As a kind friend, he selflessly shares his stories and gives suggestions, especially in this hard time that every graduates are struggling with the worldwide pandemics.

The author gratefully acknowledges the support for this work offered by Semiconductor Research Corporation. Any views and conclusions contained herein are those of the author, and do not necessarily represent the official positions, express or implied, of the funders.

Finally, I would like to thank my parents for giving me life and I enjoy it so far very much. I would also like to thank my wife Sabrina for bringing beautiful love into life.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In this dissertation, Chapter I will provide an overview of *Edge Intelligence* (EI), including a definition, a design landscape, a generic EI design methodology and reviews of state-of-art research/industrial EI demonstrations. Then, Chapter II-V will discuss several EI works in details to cover various EI design challenges/opportunities. Data-aware algorithm for EI will be discussed in Chapter II with an example of data-fusion-based edge occupancy detection for HVAC control. The fusion algorithm enables 5× detection error deduction and 3× sensor lifetime expansion while offering 26% energy savings for HVAC system in various environment through accurate occupancy detection. In Chapter III, context-aware wireless sensor control will be discussed through a self-optimizing wireless video surveillance camera platform. The self-optimizing control strategy systematically enables 4.3× energy reduction per frame compared with baseline designs. Further, a continued self-optimizing wireless image processing system-on-chip (SoC) work will be discussed in Chapter IV. This work is going to demonstrate silicon effectiveness with respect to energy, latency and area through state-of-art circuit techniques and full computation, communication and control on-chip integration. The test chip features (1) a 1.05TOPS/W (peak) programmable DNN image processor, (2) a 768pJ/b digitally-adaptive transceiver and (3) 0.59pJ/MAC actor-critic controller for efficiently controlling computation, communication blocks separately as well as jointly. In Chapter V, distributed intelligence and its hardware design considerations will be discussed via a swarm robotic application-specific-integrated-circuit (ASIC).The test chip supports both model-based and learning-based algorithm and the hybrid-digital-mixed-signal computation enables excellent scalability with number of robotic agents. A 1.1-9.1 TOPS/W efficiency is measured across various swarm size and computation precision. Finally, conclusions will be drawn in Chapter VI with both insights gained from previous chapters as well as discussions on promising future EI research directions.

CHAPTER 1

INTRODUCTION

From manually collecting and processing real-world data, to cloud-based *Internet of Things*, the evolution of technology has fundamentally changed the way human interface the real world. When we look at the trend, we are essentially trying to perceive, interpret and respond to our living surroundings more intelligently. In pursuing this goal, we have come to an era of cloud-based *Internet of Things*: *IoT* sensors collect data and transmit to the back-end cloud servers, and servers process the raw data for further action.

Despite extensive research efforts devoted to and significant societal achievements driven by such a paradigm, numerous bottlenecks inherent in cloud-IoT system have urged us to look for an alternative that can push the advances further. Firstly, wireless data has increased tremendously that creates significant transmission workload for the network. This rapid data growth has resulted from both an exponential growth of *IoT* sensors deployed in the real-word and also high dimensional data demanded by applications. These trends are manifested in Fig. 1.1. Further, the consequent heavy communication for individual IoT

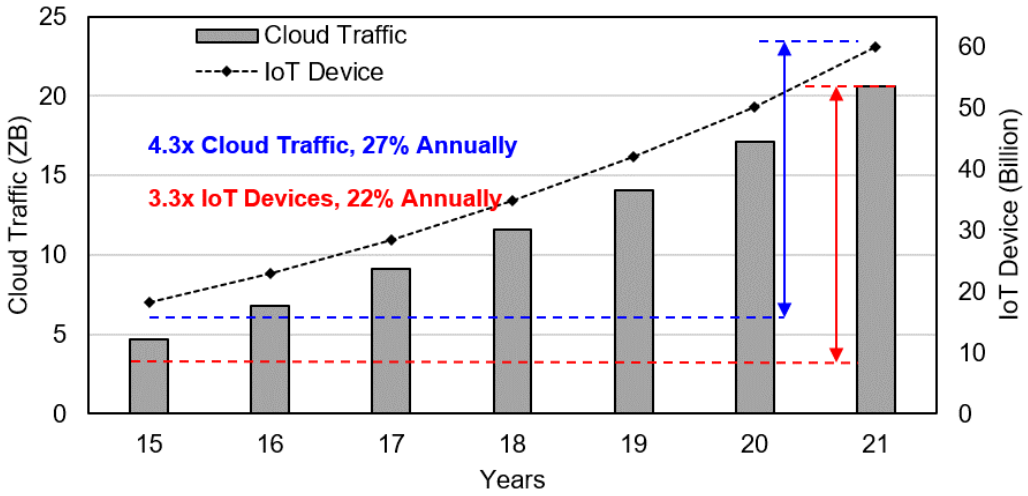


Figure 1.1: Cloud traffic and IoT device analysis [1, 2].

node brings heavy energy burden on wireless devices which either results in degraded quality of service (QoS), or increased form-factor making them less preferable. Nevertheless, due to network leaks in recent years, security/privacy concerns also prevent us from fully trusting the cloud service.

To mitigate aforementioned problems inherent in cloud-based solutions, there have been significant demands to enable *edge intelligence* (EI) paradigm for IoT development. Instead of simple "sample-transit" procedure in a centralized mannaer, EI tends to bring intelligence closer to the IoT devices that directly connect us with the real world targeting a more responsive, private and efficient IoT service. In this Chapter, I will first introduce EI and its design challenges. Then the EI design landscape and methodology will be discussed. Then, I will present a literature survey on state-of-art EI designs. Finally, I will provide an overview of the dissertation.

1.1 Edge Intelligence

"Edge intelligence" (EI) refers to the ability to empower resource-constrained edge devices at the source of the data, using advanced devices, circuits, architecture, algorithms, and

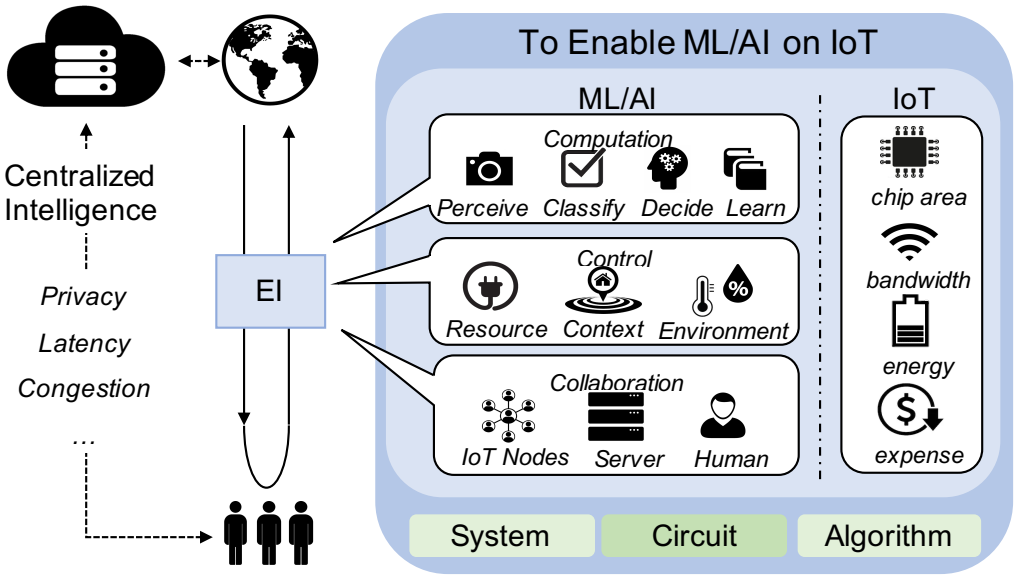


Figure 1.2: Edge intelligence overview.

control techniques to enhance the ability of data to be transformed into information (shown in Fig. 1.2). In particular, EI features machine learning (*ML*) and artificial intelligence (*AI*) applications on the IoT nodes. These edge ML/AI usages not only include computation for perception, classification, decision making and learning, but also smart control and collaboration in a dynamic wireless network. Enabling ML/AI on edge nodes will greatly enhance the smartness of the network. This improvement is reflected in many aspects.

1. EI improves IoT service efficiency. Compared to the centralized cloud-IoT paradigm that introduces enormous communication workload, EI greatly reduces the amount of data transmission via edge computation. On one hand, such a scheme minimizes network congestion hazard introduced by immense IoT devices in the network and largely mitigates the the ever-increasing communication latency problem; on the other hand, system-level energy efficiency is expected to improve by trading power-exhaustive communication with edge computation. The energy advantage is illustrated in Fig. 1.3. Processing 1 bit of data in the worst case in practice, consumes $10^4\times$ less energy than transmitting the same bit.

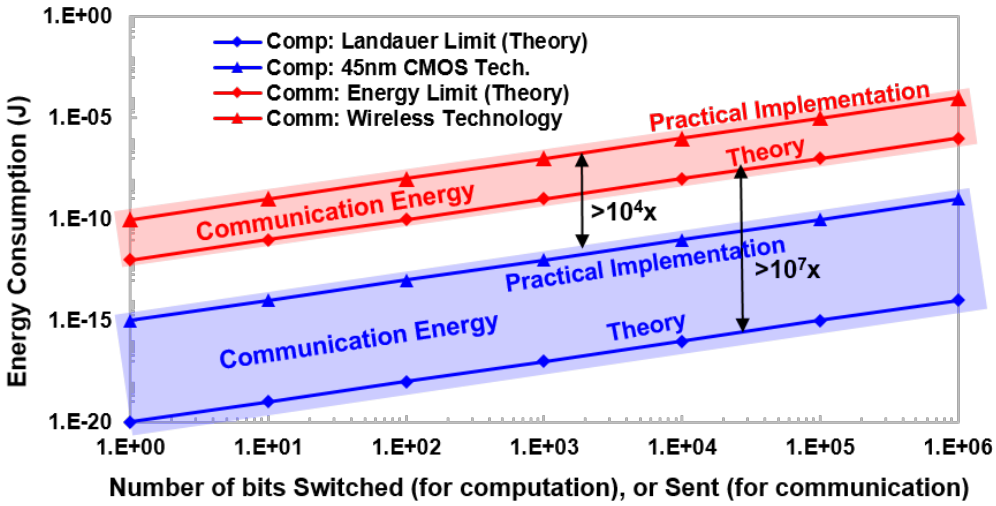


Figure 1.3: Computation vs. communication energy analysis [3]

2. EI improves IoT adaptability to dynamic environment. Embedding intelligence on end devices will help IoT system gain the capability to react to the environment in a responsive and smart manner. This allows hundreds of millions of edge devices to be dynamically controlled and self-optimized.

3. EI has inherent advantages in privacy and security. Embedded edge computation will greatly minimize information leakage risk. Needless to say that encryption essentially requires processing capability on the end devices which further enhances security.

4. EI has the potential to solve complex problems through distributed intelligence. When each edge device has a certain amount of intelligent resources and interacts with each other to form a smart cluster, they can potentially perform complex data processing tasks.

However, due to the constrained resources of IoT nodes, such as limited chip area, available bandwidth, energy storage and manufacture budget, enabling ML/AI on IoT nodes is challenging. On one hand, we have witnessed the slow demise of Dennard's scaling and Moore's Law. This has further led to slowing down of the processor performance improvement and flattening of the cost per transistor. This trend exposes the vulnerability

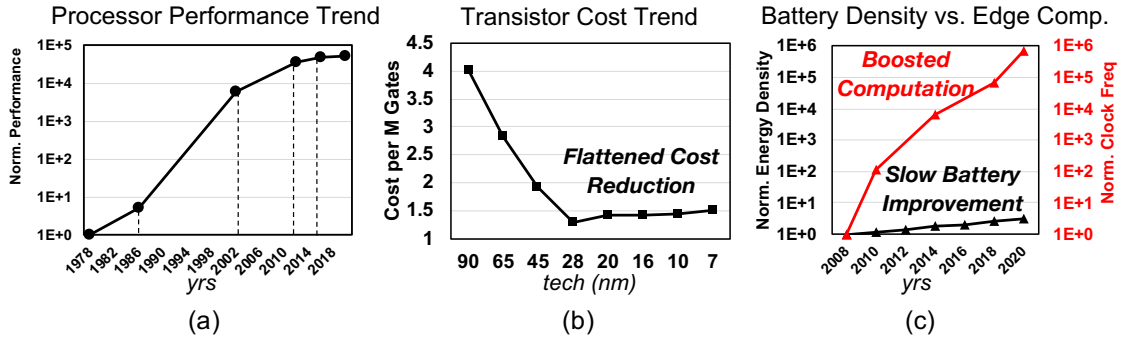


Figure 1.4: (a) processor performance scaling [4], (b) transistor cost scaling [5] and (c) battery density improvement [6] vs. edge computation required clock frequency in order to process in real time [7]

of the road map for the microprocessor performance improvement, as technology scaling will no longer be able to provide a straight-forward solution to sustain performance growth (Fig. 1.4.(a-b)). Further, the limited energy resources constrain the potential applications of edge devices. Although there has been significant improvement on battery capacity and energy-density (around 3× within 15 years), the rapidly evolving demand for complex workloads, from environmental sensing to high-definition video processing in surveillance applications, has motivated us to explore fundamental innovations in computational hardware. The real-time computation requirements for typical IoT applications in the past 10 years in terms of clock frequency are compared with the battery capacity improvement trend in Fig. 1.4.(c).

1.2 EI Design Landscape, FoM and Design Methodology

EI design is an emerging research discipline that requires systematic investigation. First of all, what are major research fields in EI landscape need to be comprehensively defined. At the same time, a figure-of-merit (FoM) has to be proposed to facilitate EI design/research evaluation procedure. Finally, a generic EI design methodology is highly desired to shed light on important design considerations and potential solutions.

1.2.1 EI Design Examples

To explore EI landscape, FoM and design methodology, the author has extensively investigated various EI research topics, including EI algorithm, control scheme, hardware integration, distributed intelligence and so on . And further, beyond theoretical/simulated efforts,to facilitate real-world performance and efficiency evaluations, the author demonstrated actual physical platforms (shown in Fig. 1.5) including: (1) a wireless self-optimizing video surveillance camera prototype; (2) a wireless HVAC data-fusion occupancy detection sensor and network; (3) a self-powered camera with compress-domain gesture-triggered wake-up; (4) a swarm robotic ASIC; (5) a wireless image processing SoC with computation-

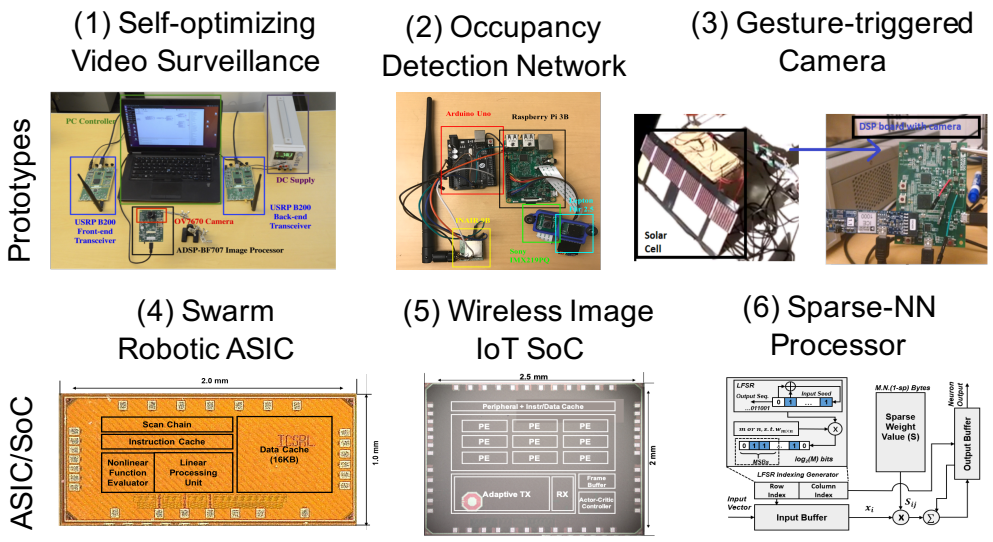


Figure 1.5: Example EI works.

communication trade-off; (6) LFSR-based DNN pruning ASIC. These works have broadly investigated research topics of IoT system, ML/AI algorithm, computer architecture, digital/mixed-signal/analog circuit design and so on. They aim at maximizing intelligence on resource-constrained devices across design layers.

1.2.2 EI Design Landscape and FoM

Despite the fact that these preliminary works covered a broad range of research topics/ML applications/design layers, like most EI works, they fall into one or more design fields in the EI landscape shown in Fig. 1.6. At the core is the capability to implement computation, especially ever-evolving ML/AI, on edge devices. It targets efficient computation through innovations in circuit techniques (example 4-5), computer architecture (example 5) or algorithm design (example 2, 6). At the same time, smart control (example 1,2,4,5) and efficient data exchange (example 1,2,5) are both crucial for the IoT to handle dynamic environment and reduce communication cost. Finally, seamless solid-state system integration of discrete components determines EI platforms' ultimate performance in the real-world (example 4-6). Although EI design demands extensive efforts from various design fields, the goal is

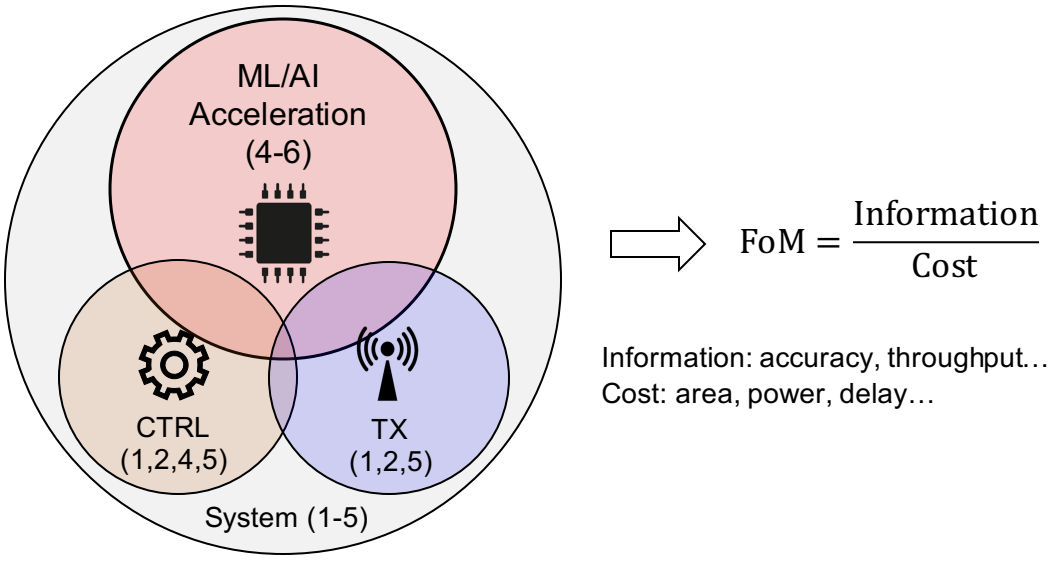


Figure 1.6: EI design landscape and figure-of-merit.

to efficiently derive ”actionable intellegence” under severe resource constraints. Or we can represent this target as a figure-of-merit (FoM): *intelligence per cost*. ”Intelligence” may be in form of accuracy, throughput and etc., and cost may be in form of area, power and etc. The EI landscape and FoM have provided a comprehensive view of EI design/research as well an evaluation criteria.

1.2.3 Conventional IoT Design Methodology

Traditional IoT devices are designed and implemented with general-purpose digital processors together with discrete peripherals, such as controller and transceiver. By evaluating EI FoM, we have found that, although they feature fast prototyping as a proof-of-concept, their efficiencies are far below our expectations for real-time ultra-low power edge devices. The details will be covered in Chapter II and III. To understand the cause of inefficiency, we need to first look at the conventional IoT design scheme described in Fig. 1.7.

Data is first sampled or produced by discrete peripherals from sensor, radio frequency (RF) modules and controllers. Then, the data is fed into digital computation system. The data will first be quantized in voltage domain, and then binary data will go through Boolean combination logic. The logic needs to be augmented with significant amount of memory

(cache, scratchpad, registers) with synchronous movement across the logic-memory boundary. At the architectural level, usually Von-Neumann architecture is adopted, where data storage (memory) and computation (arithmetic logic unit) are separated and they are controlled by a central controller. Finally, general-purpose compiler will optimize instruction execution to support universal applications.

Though it is a successful design scheme in most scenarios, but in the case of EI, it is incompatible due to the lack of efficiency caused by the various layers of abstraction from the source of physical data to the final information. Energy loss occurs when data go through each layer of the abstraction. (1) Data converters are required to transfer all physical data representations into digital voltage signals. Both information loss and ADC overhead introduces energy expenditure. (2) During Boolean operations, both dynamic

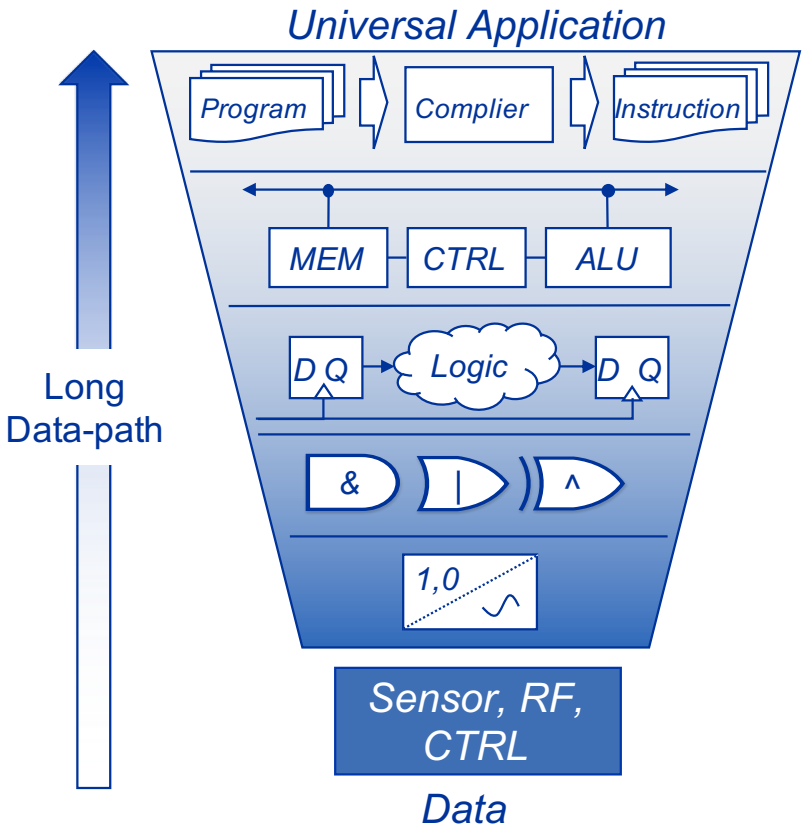


Figure 1.7: Conventional IoT design methodology.

and static power are consumed to make sure computation/storage are correct and signal is preserved rail-to-rail. (3) For the system to run in a synchronized manner, clock signal together with the large clock tree and buffers have to constantly run; thus introducing significant power consumption. (4) Separate memory and computation architecture results in significant data movement cost, especially in the era of ML/AI where large models need to constantly accessed. To push efficiency to extreme and support ML/AI on edge devices, we have to flatten the levels of abstraction bringing processing closer to the data and eliminating unwanted data-conversions and optimizing for the minimum amount of bit resolution.

1.2.4 Proposed EI Design Methodology

To flatten the design hierarchy, the author has proposed a generic EI design methodology as shown in Fig. 1.8.

1. **Context-aware Integrated System**: the system integration has to take into account the physical representation of data as well as the device's environmental context. It should be open to any physical data encoding scheme to reduce data conversion overhead, and it should adapt to the dynamics of the environmental conditions.

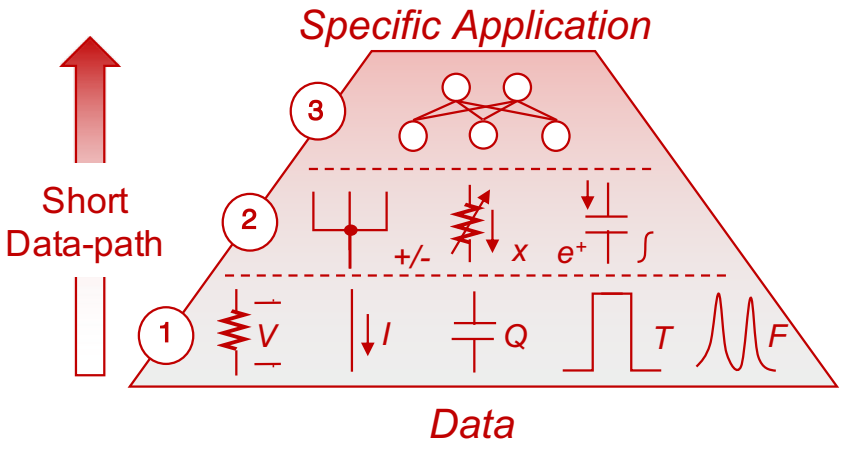


Figure 1.8: Proposed EI design methodology.

2. **Algorithm-aware Hardware**: the computational circuit design should be aware of the algorithm it implements and optimizes for the application it supports. By taking into account the data encoding scheme, any physical computation scheme can be adopted to maximize a target cost function, e.g., latency or energy-efficiency.

3. **Hardware-aware Algorithm**: the algorithm design should be aware of the hardware constraints and available hardware opportunities. We need to investigate hardware-friendly algorithms and mathematical operations that can be incorporated into the design flow.

Through proposed design methodology, the design hierarchy is flattened as much as possible. This will provide a scheme to trade-off the universality of digital microprocessors for the target FoM in EI . The details will be discussed from Chapter II to Chapter IV.

1.3 Literature Survey

As discussed in the previous sections, EI requires knowledge and innovations in various design fields. In this section, we are going to discuss state-of-art academic/industrial designs in fields of 1) data acquisition, 2) edge computation, 3) wireless communication and 4) system integration.

Upon data acquisition, researchers have been looking for opportunities in improving the capability of performing sensor data analytics while extending the lifetime of the sensors. By incorporating limited computation to the traditional sensor, the sensor can pre-process raw data to either facilitate further computation or extract useful information for optimized operation of the overall system. In computer vision field, for example, pre-processing unit for motion detection, binarization and gesture recognition [8, 9, 10] been proposed for efficient in-camera analytics. At the same time, researchers have also worked on self-powered camera sensors which convert the incident light into electrical energy to provide an in-situ energy source [11]. Another important data acquisition research field is acoustic

sensing and, in particular, ultra-low power (ULP) always-on voice activity detection (VAD) is gaining attention as an enabling technology for IoT platforms. Various silicon chip sets have been presented to enhance power efficiency, programmability and context-awareness of VAD [12, 13, 14]. Besides efforts to enhance data acquisition performance by single sensor, researchers have also investigated multi-sensors for application optimization. While a high-performance single sensor is usually both costly and power-hungery, data fusion of low-resolution sensors with orthogonal information becomes an promising alternative. [15] presents a skin-disease diagnosis system with both optical and electrical dual tomographic imaging. [16] deals with detection of occupancy in a room from various ambient sources like temperature, humidity, light, and CO2 to leverage HVAC control in real time.

At the core of EI, the ability to handle computation-intensive task on the edge hardware is a major challenge. With slow-down of Moore's law as well as the reducing opportunities of scaling in digital VLSI, analog and mixed-signal circuit innovations are being actively explored. These innovations include dedicated ultra-low-power, moderate precision mixed-signal/analog computational block, but also architectures to improve data movement/computation. For example, [17] presented an SoC that performs continuous-time hybrid approximate computation,in which both analog and digital signals are functions of continuous time. [18] demonstrated a matrix multiplying ADC to enable feature extraction and classification with data conversion, mitigating the need for further computation. Similarly, [19] built a switched-capacitor matrix multiplier with co-designed bitline-less memory to reduce A/D conversion rate and improve MAC computation energy efficiency. A novel spike-based SLAM accelerator has been presented in [20]. [21] features an energy-efficient switched-capacitor (SC) neuron that addresses energy challenge, employing a 1024-bit thermometer-coded capacitive digital-to-analog converter (CDAC) section for summing point-wise-products of CNN filter weights and activation and a 9-bit binary weighted section for adding the filter bias. Meanwhile, with ever-increasing data-centric computation, people are also looking for alternatives for von-Neuman architecture, where

data storage and computation are tightly coupled. [22] demonstrates a 7b energy efficient SRAM with embedded convolution computation for CNN-based machine learning applications. [23] proposed a 'sandwich' architecture of in-memory binary weight network (BWN) to blend feature and partial-weight memory with a computing circuit together that achieves significantly less data access. From a larger scope, the spatio-data-correlation has been investigated and utilized to improve system preformance as in [24]. [25] presented a nonvolatile compute-in-memory ReRAM macro for binary DNN AI edge processors for low power feature and fast IO accesses. Nevertheless, novel devices, especially energy-efficient, low-cost and high data-rate non-volatile memory together with compatible computation architecture are widely researched. [26] demonstrates Ferroelectric FET Analog Synapse for Acceleration of Deep Neural Network Training. [27] provides a 3D-flash memory with improved area capacity. [28] built STT based RAM for high-yield, high performance and high-endurance.

Towards a fully connected internet of things, innovations in wireless data communication is crucial in maximizing datarate while minimizing latency, energy-per-bit (EPB) and bit-error-rate (BER). On the circuit level, researchers mainly target improved RF block metrics, such as improved transmission resolution at low loss [29] or obtaining high output power while maintaining efficiency [30]. At the same time, the community is also exploring SoC-level RF solutions, such as [31], to push throughput limits and enable 5G networks and beyond. At the same time, as communication units are usually designed to meet minimum requirement at worst-case, energy/performance are not optimal in a dynamic environment. In this background, cognitive radio where RF is optimally controlled in a context-aware manner is widely investigated. [32] advocates the use of Reinforcement Learning (RL) incontext-aware and intelligent Dynamic Channel Selection (DCS) scheme to enhance QoS in Cognitive Radio (CR) networks. [33] demonstrates orthogonal tuning knobs using an inductorless LNA which has 14dB Gain tuning range and 30 dB OIP3 tuning range with power consumption goes down by 20. [34] develops a multidimensional

adaptive power management approach that optimally trades-off power versus performance across temporally changing operating conditions by concurrently tuning control parameters in the RF and digital baseband components of the wireless receiver. [35] demonstrates a real time BER vs. power consumption modulation of RF front-end devices in MIMO system. Beyond circuit and context-aware control, communication schemes for dedicated scenarios are also important for optimized performance. [36] demonstrates characterization of human body communication for ultra-low power high-accuracy medical applications. [37] reports results from wireless chip-to-chip communication experiments with 16 bit words pass from one chip to another in parallel without detectable error at 1.35 billion data items per second for a total data rate of 21.6 Gigabits per second.

On top of efforts in optimizing each functional block of EI (data acquisition, edge computation and wireless communication), efficient system integration is also challenging and provides the ultimate evaluation metrics for EI hardware design. [38] presented a complete "edge-gateway-cloud" IoT system prototype for an example application that highlights the key advanced capabilities of the cm-scale, self-powered, intelligent and secure mote hardware platform at the edge. In [39], Intel showed another wireless sensor node (WSN) that integrates near-threshold voltage (NTV) 32-bit Intel Architecture (IA) microcontroller (MCU) in 14nm tri-gate CMOS, along with solar cell, energy harvester, flash memory, sensors and Bluetooth Low Energy (BLE) radio, to enable always-on always-sensing (AOAS) and advanced edge computing capabilities in Internet-of-Things (IoT) systems. [40] demonstrated a low-power Robot SoC in 22nm CMOS that is integrated in the cm-scale minibot platform along with audiovisual and motion sensors, battery, low-power wireless communication and motion actuator components. [41] presents a single chip VLSI architecture of wireless image sensor node, which is constituted by an enhanced embedded 8051 microcontroller, a CMOS camera interface and hardware accelerators.

As a broad topic and emerging area, it is challenging for individual researchers to demonstrate a complete study of EI. Instead the field continues to advance through collabo-

rations. During the literature study, the author has found several informative survey papers providing insightful knowledge from various perspective. [42] provided an overview of technologies associated with IoT in embedded systems' landscape. They have investigated essential technologies for development of IoT systems, existing trends, and its distinguishing properties. By discussing the key characteristics, main application domains, and major research issues in IoT, this paper provides a comprehensive IoT perspective for embedded system design. On the other end, from an an industry and government perspective, [43] provided insight to the problem of intelligence resource constrained IoT nodes and presented the vision of the future and important technologies that might play a strong role in enabling the vision of trillion smart connected sensors. Finally, [44] provided an academic perspective of the problem, starting with a survey of recent advances in intelligent sensing, computation, communication, and energy management for resource-constrained IoT sensor nodes leading to future outlook and needs.

1.4 Dissertation Overview

In the following Chapters, the author will discuss algorithm design, control strategy, system integration as well as distributed intelligence via dedicated example works. In Chapter II, a fusion-based occupancy detection algorithm, together with an HVAC occupancy detection wireless camera system will be discussed. In Chapter III, an online EI computation-communication trade-off control scheme is introduced with a wireless video surveillance camera. Continuing further in the same light, Chapter IV describes an SoC implementation of computation-communication trade-off control strategy in wireless image processing applications. The silicon effectiveness in energy, latency and area will be fully investigated. After discussing single-agent EI algorithm, control and system design, Chapter V is going to discuss multi-agent distributed EI through a unified swarm robotic ASIC. In particular, this Chapter will address hardware requirements to meet scalability issues in multi-agent learning scenarios. Finally, the author will draw conclusion in Chapter VI.

CHAPTER 2
EI FOR DATA-FUSION-BASED OCCUPANCY DETECTION IN HVAC CONTROL

IoT devices are widely used to sense environment and provide information to assist system-level decision makings. As a result, the accuracy in data acquisition and processing is critical for the whole system to work properly. However, the constrained resource has made accuracy enhancement via complex models more challenging. It is highly desirable that EI algorithm design can fully utilize all available data, reduce model dimensionality and enhance performance while meeting energy/latency budget. This chapter discusses a data-fusion-based occupancy detection algorithm for HVAC control to provide an EI algorithm design example on how to augment information per cost via data-aware algorithmic optimization. This chapter is a slightly modified version of "Smart sensing for HVAC control: Collaborative intelligence in optical and IR cameras" published in IEEE Transactions on Industrial Electronics with the dissertation author as the primary author.

2.1 Introduction

HVAC provides a comfortable climate controlled environment at home and work. However, trillions of kWhs of electrical energy are consumed annually in the U.S.(Fig. 2.1.a) which accounts for more than 30% of the energy consumed in all residential and commercial buildings. $10 - 40\%$ of this electrical energy is wasted due to inefficiencies, such as unnecessary HVAC operation, over-estimated temperature set point etc. This is further exacerbated by poor and aging insulation on the walls, doors and windows. To address this challenge, different approaches have been developed. Programmable thermostats, which turn off the HVAC system when the house is expected to be vacant, is one common approach for efficient HVAC energy use. However, this approach repeatedly fails at predict-

ing the room occupancy in highly dynamic occupancy pattern [45, 46]. An alternative to a manually programmable schedule-based thermostat is the occupancy-based HVAC system which dynamically senses room occupancy and adaptively controls itself. Among these dynamic detection approaches, RFID (radio-frequency identification) tags [47] and infrared sensors are popular. RFID tags require humans to wear badges or tags in person, which is an inconvenience and not applicable to residential buildings. Infrared sensors require motion [48, 49]. On the other hand OP camera based sensors show great potential in occupancy detection [50, 51] even when the occupants are static. However, OP camera-based occupancy detection remains challenging. They either suffer from high miss rates, resulting in discomfort in the room; or high false positive rates (recognizes a non-human object as human) leading to energy wastage in a vacant area. The trade-off between miss rate and false positive rate and their impacts on occupancy-based HVAC system are illustrated in Fig 2.1(b)-(c).

To achieve high occupancy detection accuracy, thus improving occupancy-based HVAC performance, this work presents a collaborative intelligence solution via data-fusion between OP and IR camera-based sensor nodes together in a smart wireless sensor network. Collaborative intelligence is achieved *at the sensor node* as well as *among the sensor nodes* at the back-end server, which is located at the HVAC and controls the HVAC. With minimal HW/SW overhead, improved detection accuracy results in enhanced comfort, extended sensor lifetime and reduced HVAC energy wastage.

2.2 Platform description

Before going into the algorithmic details, let us discuss the experimental setup. Our prototype sensor comprises of an OP camera (IMX219PQ), an IR camera (Flir2.5), an embedded processor for image processing (Raspberry Pi), a long range radio (LoRa) (INAIR9B) and a transceiver controller (Arduino Uno). The system architecture is shown in Fig.2.2(a) and the platform hardware setup is in Fig.2.2(b). OP and IR images are captured, aligned and

processed to determine occupancy in the field of view (FoV). The sensed output, including occupancy/vacancy transient or motion vector, is transmitted to the HVAC controller, often at the basement of an office building, using a narrow bandwidth long range (LoRa) radio, which is duty cycled to prevent unnecessary energy expenditure at the sensor. It should be noted that we use LoRa in this work; however, other communication protocols, such as Wi-Fi can be used in the sensor node. The choice of the communication depends on the available infrastructure, requirements on power as well as distance over which the sensor node needs to send necessary signals. This is described in more details in the following sections.

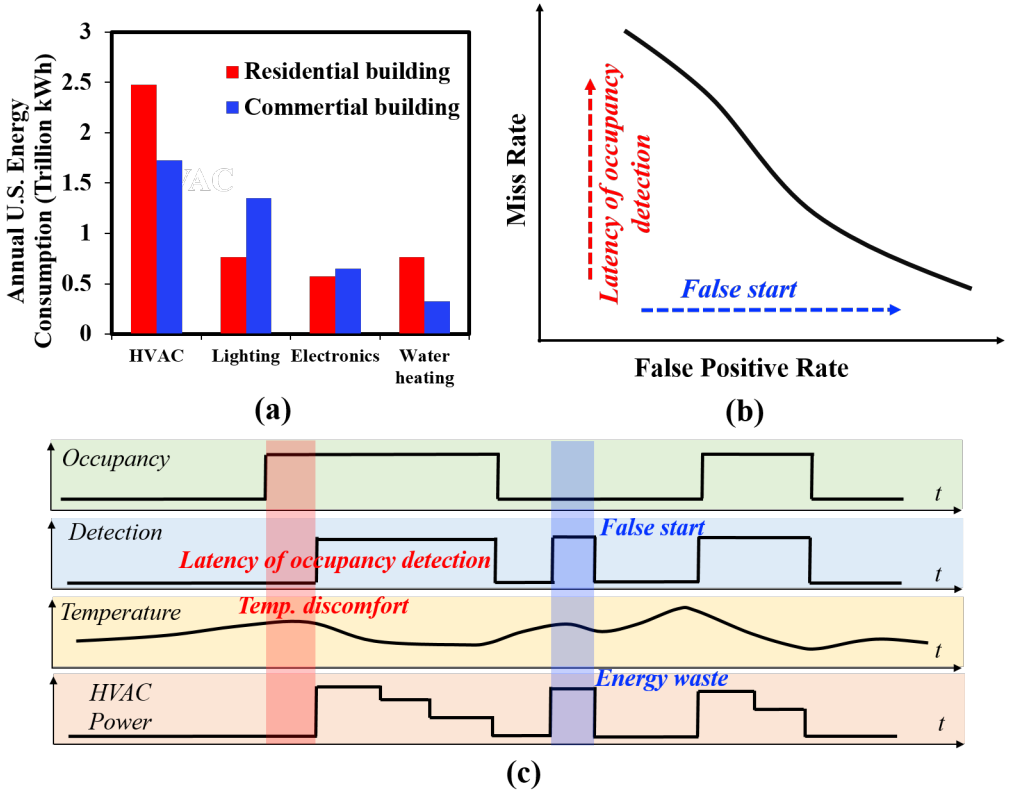


Figure 2.1: (a) Residential and commercial energy use [52]; (b) Qualitative assessment of the trade-off between miss rate and false positive rate; (c) The impact of miss/false positive on latency of occupancy detection/energy waste respectively.

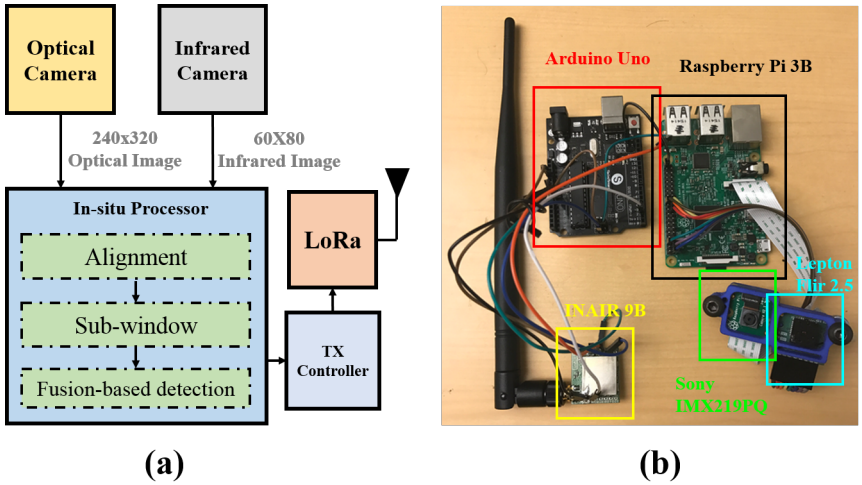


Figure 2.2: (a) System architecture of the platform prototype; (b) Hardware setup.

2.3 Occupancy detection via Collaborative Intelligence

Occupancy detection has long been investigated, and different types of special-purpose sensors have been proposed. For example, IR motion sensors, RFID, door sensors, image sensors etc. Some sensors suffer from low detection rates and some require additional devices to be worn by the occupants to assist detection. In some of these sensors, the occupants need to be in motion, or else the sensor fails to detect occupancy. Most current solutions suffer from low accuracy or longer latency of detection. Our proposed system

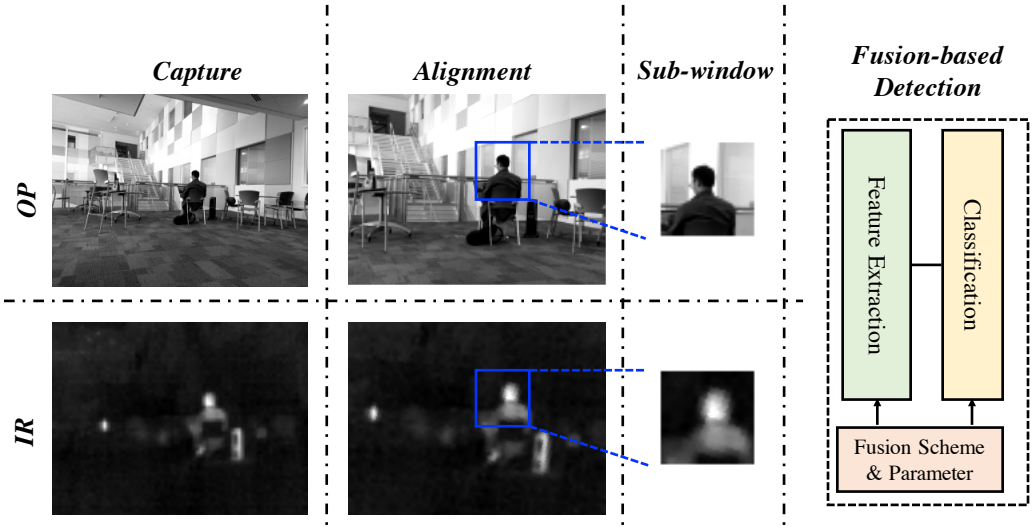


Figure 2.3: Demonstration of the algorithm.

demonstrates the simultaneous use of both OP and IR cameras. The sensor node fuses the two images, and improves detection rates while minimizing false detection.

Instead of sensing illumination intensity or color as OP cameras do, an IR camera perceives heat emissions. Occupancy detection through IR camera is a promising approach to counter the failures via OP cameras that are caused by darkness, optical foreground/background similarities and partial occlusion. However, apart from human, there are other sources that emit IR waves, such as sunlight, machines etc. which result in false detection for an IR only system. Furthermore, typical (and inexpensive) IR cameras are low resolution, which prevents efficient machine learning algorithms to detect features of a human being in the IR domain. Therefore, we combine the advantages of OP and IR detection schemes to provide accurate detection by fusing OP/IR data. The platform captures OP/IR images simultaneously, aligns them by image registration, fuses the information collected from the registered images, and determines the room's occupancy/vacancy from the fused data.

2.3.1 Overview

Camera-based occupancy detection, either OP or IR, can be further categorized as video-based and image-based. A typical video-based detection takes temporal difference [53] between frames, so a successful detection relies on the motion of objects. Therefore, video-based detection will definitely fail for static humans. The image-based approach, in contrast, depends on shapes of objects and independent of objects' motion. Thus, the proposed system applies image-based detection to handle both moving and motionless objects for realistic residential and commercial building occupancy detection.

After the OP/IR images are captured, the two images are aligned by 2-D image registration. Then, histogram of oriented gradient (HOG) feature are extracted and classified by an artificial neural network (ANN) template. This algorithm is outlined in Fig. 2.3.

Image alignment

Accurate alignment of data acquired by sensors with different characteristics is essential in data-fusion. In the proposed platform, the FoV and resolutions of OP image sensors and IR image sensors are different. Images captured from the two cameras first need to be registered as a preliminary data-fusion procedure upon deployment to cross-check regions of interest. We assume a long distance between the human and the platform (mounted on the wall near the ceiling), so that any human being can be regarded as 2-D, together with a parallel placement of the OP and IR camera sensors [54]. We apply a rigid translation and the 3-D translation can hence be decomposed into 2-D by:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = s \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \tag{2.1}$$

where a 2-D point (X,Y) in OP image is transformed to a 2-D point (X',Y') in IR image with a scaling factor s, rotation angle θ and offset $(\Delta X,\Delta Y)^T$. Since θ, $(\Delta X,\Delta Y)^T$ are recorded during installation, we only need to perform a calibration to obtain the registration matrix for every image.

Feature extraction

Feature extraction derives informative and non-redundant values to facilitate the subsequent stages to generate better classification results. It is a key stage in performing classification with high accuracy. In human detection, feature extraction is crucial to discriminate human from cluttered background. Different feature descriptors are available, including wavelets, SIFT and HOG. Among all feature extractors, Histogram of Gradient (HOG) is chosen for its excellent performance [55, 56, 57]. HOG first divides the input image matrix evenly into M$\times$N cells. Gradient angle and gradient magnitude of each pixel are computed. Each pixel within the cell votes for an orientation-based histogram channel by comparing gradient

angle with angle bins with weight of gradient magnitude. Angle bins evenly spread on $(-\pi, \pi]$ range and number of bins is N_{bin}. Then the spatially connected cells form a block of size (M-1)×(N-1) to be locally normalized to account for changes in illumination and contrast where M and N stands for number of rows and columns of cells.

Classification

Classification is the final step of detection which takes extracted feature descriptor as an input, compares it with a trained template, and outputs scores indicating the likelihood of a detection (occupied vs. unoccupied). Among different classification methods such as support vector machine (SVM), Naïve Bayes, tree etc., a three-layer artificial neural network (ANN) is selected for its improved performance in classification and linear computation cost with input size [58]. ANN is an information processing paradigm inspired by biological neural system consisting of input layer, hidden layer and an output layer. In the current design, the number of input layers is the same as the feature size N_f with hidden and output layers of sizes N_h and N_l respectively. The output score of the input feature vector is computed as follow:

$$Y(\vec{x}) = \sum_{i=1}^{N_{hl}} [\alpha_i \sum_{j=1}^{N_f} (\omega_{ij} x_j + \gamma_{ij}) + \beta_i] \tag{2.2}$$

where x_j is the j_{th} element of the input feature descriptor; and ω_{ij}, γ_{ij}, α_i and β_i are i_{th} hidden neuron weights, biases, output neuron weights and biases for j_{th} input element respectively. In the proposed design, the hidden layer neuron size is fixed to be 100. The computation cost of classification increases with the feature descriptor size. This relation will be discussed in the following section.

OP/IR database

With the rapid development of machine learning (ML) technique, a great many OP and IR data sets are generated, such as INRIA, MPII, InfAR and etc., to facilitate ML-based OP

and IR detection/recognition/classification tasks. However, a good "OP+IR" in-door human data set, where OP and IR images are captured simultaneously targeting fusion-based ML, is still not available to our knowledge. Previous work, such as [59], is only based on limited or separate data set. This has motivated us to create our own data set which (1) contains substantial pairs of positive and negative "OP+IR" pictures and (2) pictures demonstrate diverse OP/IR foregrounds and background features. The collected data set contains 3727 pairs of 40×40 8-bit gray-scale OP/IR images, 1928 positive and 1799 negative, covering foreground samples of human in different postures, clothing and so on; and background in different lighting conditions and infrared intensities. *(The database will be publicly released and is currently not linked for the blind review process).* Furthermore, to demonstrate realistic results and avoid sample testing, training data and testing data are populated separately with totally different human foreground and OP/IR backgrounds.

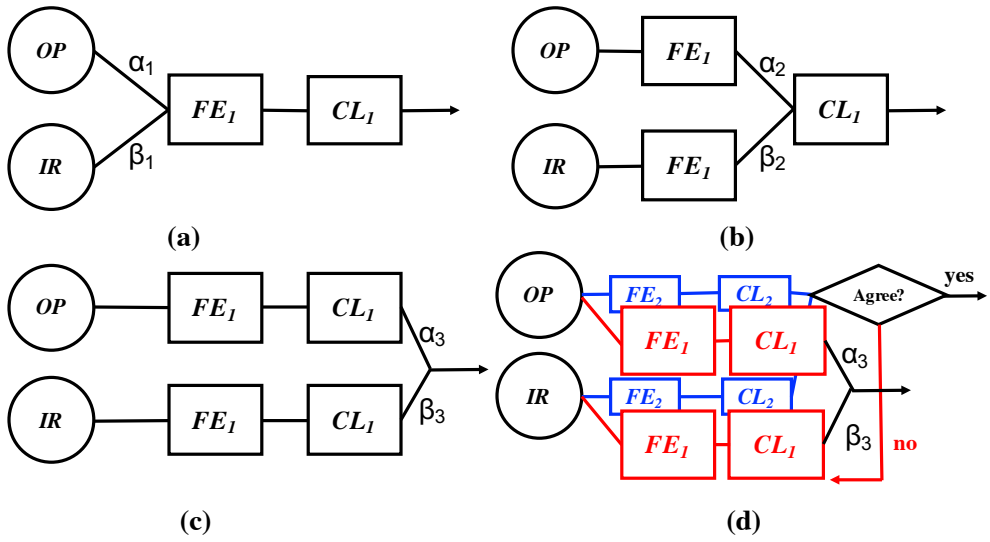
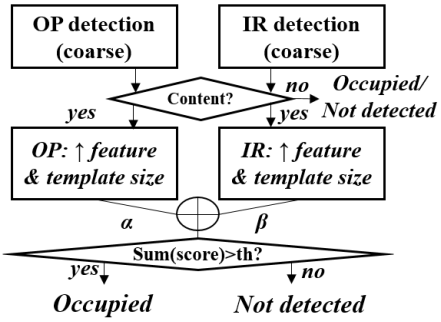


Figure 2.4: (a) Data-level fusion; (b) Feature-level fusion; (c) Decision-level fusion; and (d) The proposed collaborative, hierarchical and adaptive template (CHAT) algorithm. Here FE and CL denote feature extraction and classification respectively.

OP	IR	Decision
No	*Yes*	*CHAT*
Yes	*Yes*	*Occupied*
Yes	*No*	*CHAT*
No	*No*	*Not detected*

(a)



(b)

Figure 2.5: (a) OP/IR decision table for CHAT; (b) Flow chart of CHAT.

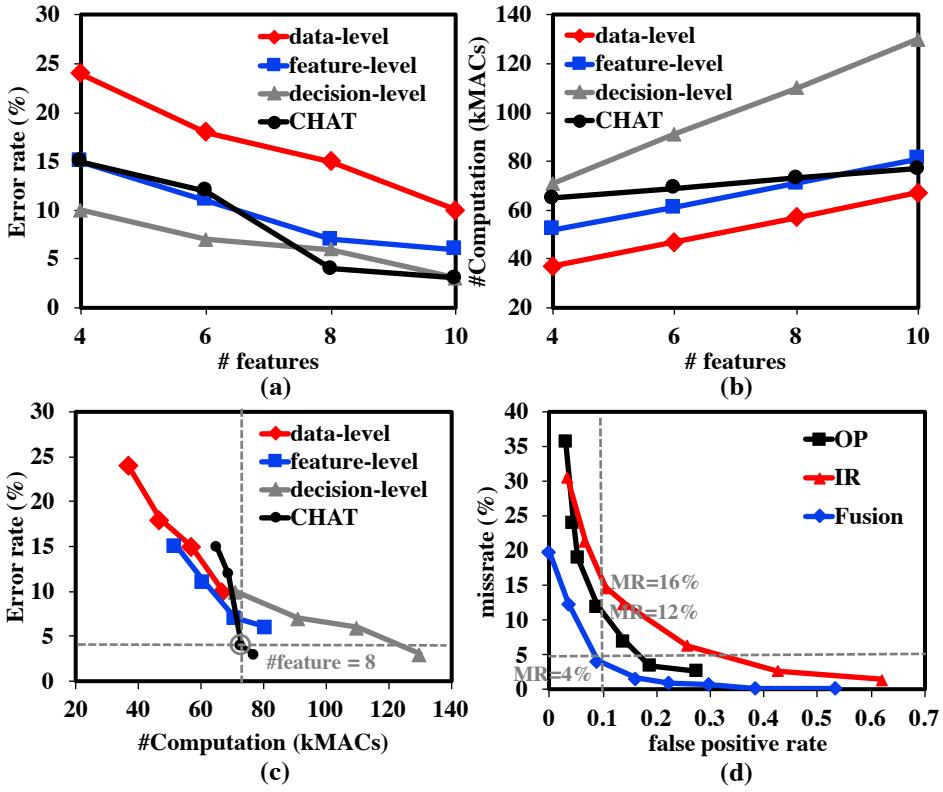


Figure 2.6: (a) Error rate vs. number of feature bins; (b) Computation load vs. number of feature bins; (c) Error rate vs. computation cost for different fusion scheme; (d) ROC of OP, IR and "OP+IR" fusion-based detection.

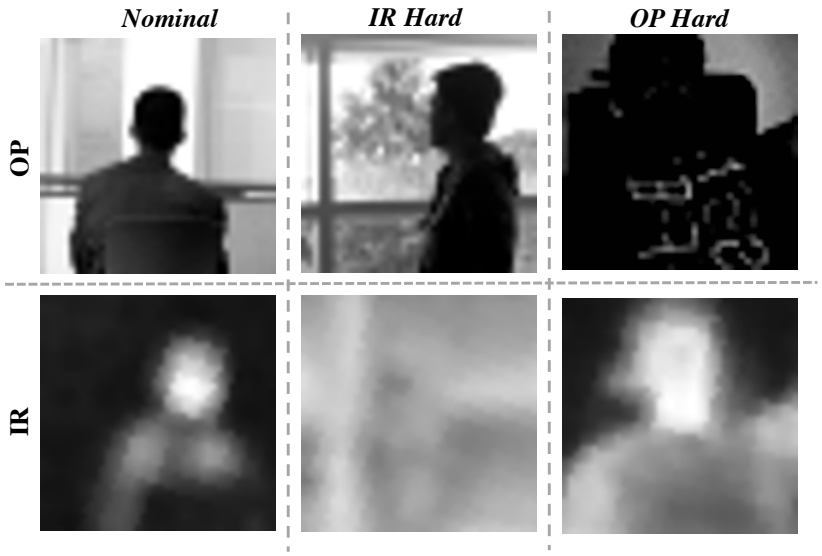


Figure 2.7: Prototypical example of nominal, IR hard and OP hard data sets.

2.3.2 Fusion-based detection

Multi-sensor data fusion is an emerging technology applied to various areas such as automated target recognition, battlefield surveillance, autonomous vehicles and so on. It combines data from multiple sensors and related information from associated databases to achieve improved accuracy than could be achieved by single sensor alone.

A key issue in developing a multi-sensor data fusion system is the question of where to accurately combine the data in the data flow. Typical schemes are data-level fusion, feature-level fusion and decision-level fusion [60, 61] as shown in Fig.2.4.(a)-(c).

In the proposed occupancy detection system, data-level fusion refers to integrating aligned OP and IR with different weights into a combined single frame, extracting HOG features from combined data and classifying the feature with the ANN template. Feature-level fusion refers to extracting HOG feature of OP/IR frames separately and concatenating the weighted two feature descriptors into one single feature descriptor for ANN template inference. Decision-level fusion refers to separate OP/IR frame evaluation and use the weighted sum of output scores to indicate human occupancy.

Apart from those traditional approaches, this work presents a novel fusion scheme with

collaborative, hierarchical and adaptive template (CHAT) as is shown in Fig.2.4.(d). A "coarse" feature descriptor is first extracted for OP and IR images and evaluated by the corresponding "coarse" ANN templates; if two sensors reach consensus, the system outputs the agreed decision, as is shown in Fig.2.5.(a), otherwise, it goes back and follow a decision-level fusion with "fine-grain" feature extraction and classification, as is shown in Fig.2.5.(b). The advantages of such an approach are (1) computation cost savings for easy detection environments using the "coarse" feature templates and classification; and (2) accuracy improvement by resolving contentions via hierarchical template adaptation.

To fully explore detection performance and computation cost of the four fusion schemes, experimental results are demonstrated in Fig.2.6. As we can observe from Fig.2.6.(a)-(b), a larger feature space helps improve detection accuracy at the cost of increased computation.

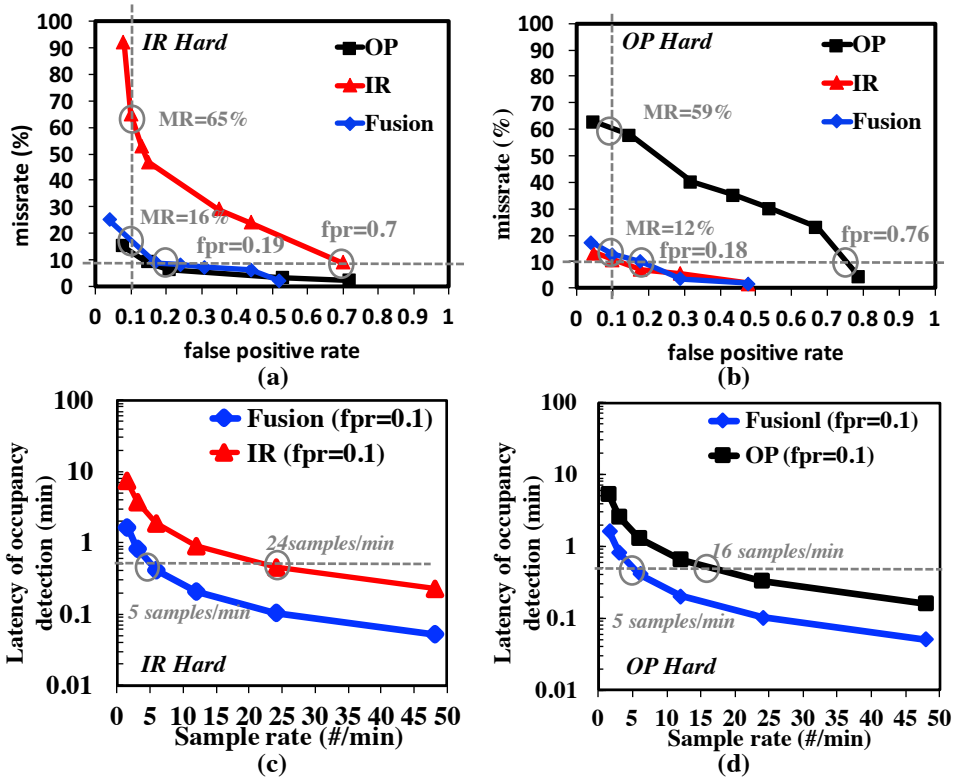


Figure 2.8: Measured (a) ROC in IR hard scenario; (b) ROC for OP hard scenario; (c) latency of occupancy detection vs. sample rate in IR hard scenario; (d) latency of occupancy detection vs. sample rate in OP hard scenario.

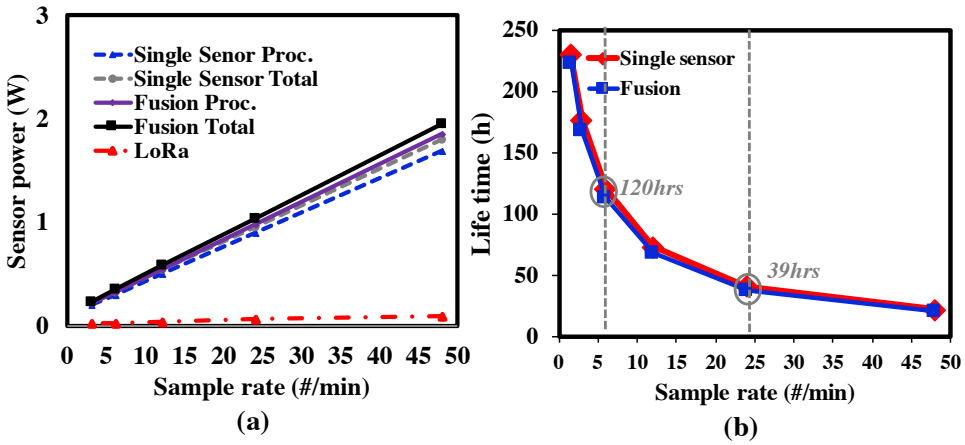


Figure 2.9: Measured (a) sensor power consumption vs. sample rate, including processing power, LoRa and total power; (b) sensor life-time vs. sample rate for single and fusion-based sensor

After feature bin size equal to or greater than 8, the advantage of CHAT in both detection performance and computation cost become apparent. In Fig.2.6(c), the maximum accuracy and minimum computation is observed. Based on the measurements above, the proposed system selects CHAT as the fusion scheme and all the fusion-based experimental results discussed in the rest of the chapter use the CHAT fusion algorithm. The receiver operating characteristic (ROC) of single OP, single IR and CHAT fusion-based detection schemes in the nominal case are demonstrated in Fig.2.6.(d). At false positive rate of 0.1, fusion-based detection achieved $3\times$ and $4\times$ miss rate reduction for single OP and IR detection.

2.3.3 OP/IR hard detection and sensor lifetime

Despite the improvement of detection performance in general, a significant benefit of applying fusion-based detection in real environment is that it maintains reasonable detection accuracy when OP and IR sensor alone will fail in extreme cases, and extend the sensor lifetime.

Fig.2.7 shows three illustrative cases of data: the nominal case, where both the OP and IR foreground show contrast against the background; the IR hard case, where the infrared background is similar to the humans in the foreground; and the OP hard case, where the

OP background is similar to the humans in the foreground. In the latter two cases, the corresponding single sensor system will show heavily degraded performance, more than 50% detection miss at fpr of 0.1, as is shown in Fig.2.8.(a)-(b).

As the proposed system is duty-cycled to save sensor energy consumption and extend the sensor lifetime, latency of occupancy detection, which is the interval between a person entering the FoV and being detected, depends largely on both the sampling rate and the miss rate for a certain fpr as is shown in Fig.2.8.(c)-(d). Here fpr is maintained at 0.1 for both the IR and OP hard cases: the latency of occupancy detection is reduced at higher sampling rate. The proposed system detects objects more quickly than the corresponding baseline single-sensor designs owing to its lower miss rate. To maintain a maximum target latency of occupancy detection of 30 seconds, fusion-based sensor platform can sample more slowly and it thus prolongs the sensor lifetime as is shown in Fig.2.9, despite the energy overhead brought by an extra sensor.

2.4 Smart sensor network

In both residential or commercial buildings, more than one sensor is required to cover the whole HVAC zone. These front-end sensors will form a wireless network and periodically transmit sampled data to the back-end HVAC controller. In designing such a network, the maximum number of nodes, maximum range of the network, average sensor power/energy as well as system level detection performance are primary concerns. In the proposed system, the intelligence of the front-end sensor is complemented with a network level inter-dependent wake-up mechanism which optimizes the target design metrics. In this work, we use LoRa as the communication protocol. However, other protocols such as Wi-Fi can be used as well depending on the availability.

2.4.1 Intelligent LoRa front-end

In conventional wireless sensor networks, front-end sensors usually follow a centralized "sense-transmit" working scheme: raw data are captured and transmitted directly to the back-end data-center without any in-situ intelligence. However, for camera-based data-intensive application, such communication strategy not only results in high communication energy and hence low battery life of the sensor node, but also network congestion producing severe quality of service (QoS) degradation in the form of queueing delay at best, and packet loss or blocking of new connections in the worst case [62, 63, 64, 65].

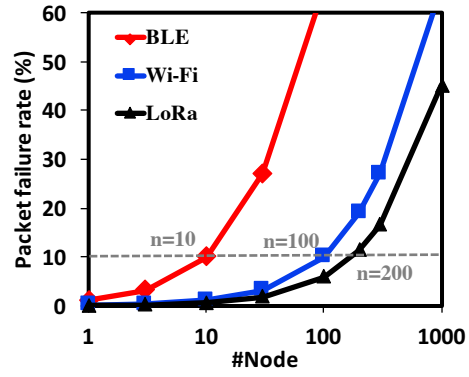
To address these issues, the proposed system is equipped with in-sensor data processing capability and notifies the back-end controller (located at the HVAC control) when an area is occupied through low-bandwidth/low-power long range radio (LoRa). Here we numerically compare wireless front-end of conventional "sense-transmit (BLE, Wi-Fi)" strategy with the proposed intelligent LoRa wireless scheme.

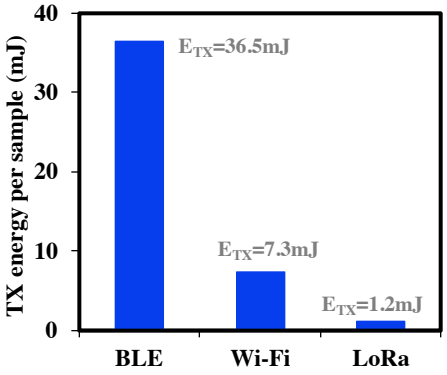
Packet arrival is modeled as a Poisson process [66] and compared with "sense-transmit (BLE, Wi-Fi)" whose raw data are always transmitted and processed at the back-end. In comparison, de-centralized "embedded computation + LoRa" suffer from less packet failures in wireless networks with a large number of nodes, as is shown in Fig.2.10.(a). For example, at 10% packet failure rate, The maximum number of sensors in LoRa network is 200, much larger than the estimated 10(100) of BLE(Wi-Fi) counter-part. Further the Wi-Fi network is heavily utilized for data transmission, and its use for HVAC control will further exacerbate the network congestion.

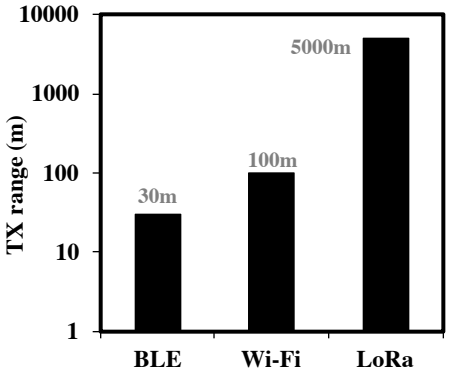
An important factor for wireless network is the average transmission energy per sample for individual sensor front-end, estimated as:

$$E_{tx} = \frac{p_{tx}\tau}{N_s} \tag{2.3}$$

where E_{tx} stands for transmission energy per sample and p_{tx} is the active transmission







	BLE	Wi-Fi	LoRa
Packet length (kb)	700	700	0.08
TX/RX bandwidth (kbps)	2000	20000	2.5
τ (s)	0.35	0.035	1
R_S (sample/s)	0.05	0.05	0.05
N_S (pkt/sample)	1	1	0.021
P_{TX} (mW @ 4dBm)	104.3	209	59.4

(d)

Figure 2.10: Estimated (a) packet failure rate vs. number of node in a wireless network for BLE, Wi-Fi and LoRa respectively; (b) transmission energy consumption per sample; (c) transmission range; (d) parameter table.

power. Here we assume transmission time interval is equal to minimum receiver window, τ. From Fig.2.10.(b) we observe that the LoRa based sensor consumes the least amount of battery energy in our wireless networks for HVAC control.

Apart from the network capacity and transmission energy, transmission range is also important in controlling large HVAC regions, especially in warehouses, and large office buildings. As is shown in Fig.2.10.(c), the range of LoRa radios outperform BLE and Wi-Fi by at least 10×.The system parameters are listed in the table of Fig.2.10.(d).

As we have seen, an advantages of LoRa is the long range over which communication can happen, which is relevant to large commercial buildings or warehouses. Since the in-sensor processor will reduce the data volume that needs to be transmitted to the back-end,

a narrow-band protocol such as LoRa is an excellent choice. However, Wi-Fi can also be used if available.

In our current experimental setup, the Raspberry PI includes an integrated Wi-Fi radio. We have also enabled an Aduino based LoRa radio that interfaces with the Raspberry PI. An added advantage of using LoRa in the current set-up is the ability to enable fine grain power management (turning on and off the radio) through the Arduino board – thereby reducing power. More advanced designs, including ASICs may further improve power management on the radio by enabling fast wake-up and sleep; but it is outside the scope of this work.

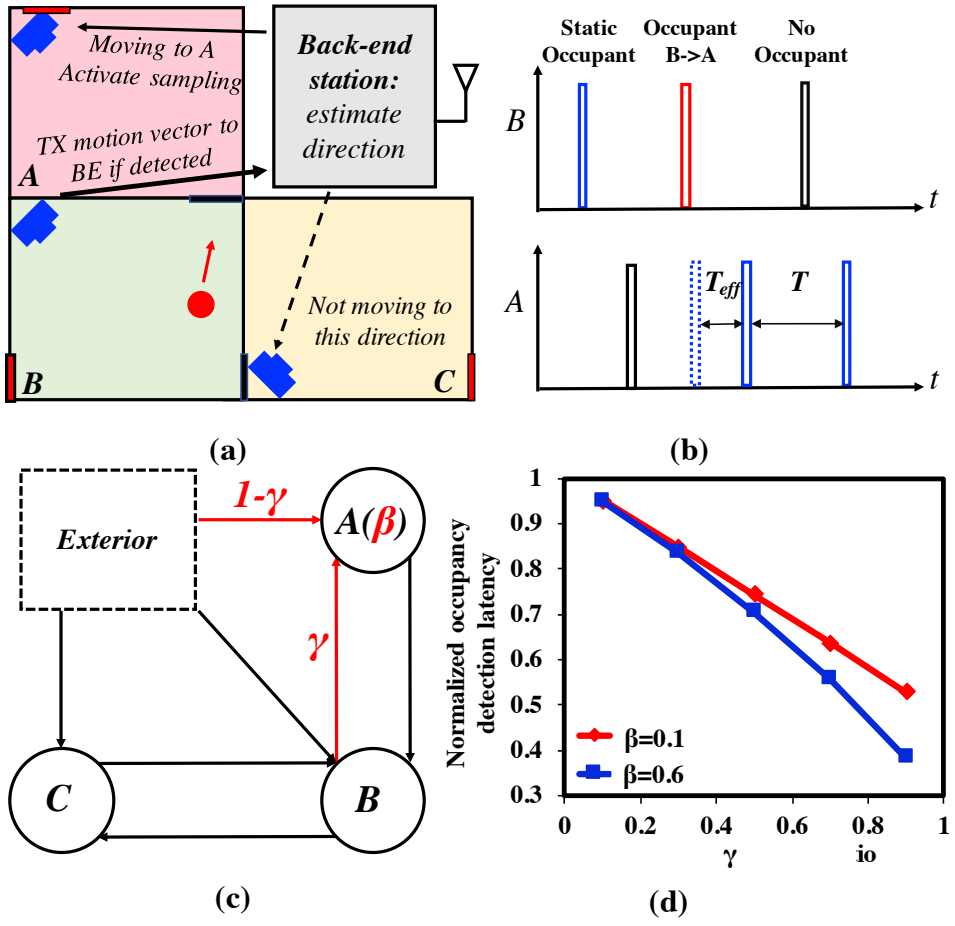


Figure 2.11: Illustrative representation of (a) a simple sensor network with inter-dependency; (b) demonstration of event-driven sampling; (c) network topology; (d) estimated latency of occupancy detection reduced within the collaborative network.

2.4.2 Collaborative dynamic network control

When detection accuracy is fixed after the employment of the sensor network, minimizing latency of occupancy detection depends on reducing the sample interval (i.e., the number of OP and IR images captured per second), T. However, a high sampling rate will lead to severe sensor energy expenditure and limited sensor lifetime. And it is also noted that the occupancy of a particular region in a building is dependent on its neighboring regions. For example, consider a typical floor-plan of a building with three rooms, A, B and C. The occupancy of room A is dependent on room B if a door between A and B is available and people can walk from B to A as is shown in Fig.2.11.(a), and vice versa. This motivates the proposed dynamic HVAC control strategy targeting minimized latency of occupancy detection based on a collaborative scheme among neighboring HVAC sections.

Consider a network of sensors deployed as shown in Fig.2.11.(a). The sensor node at B estimates the presence of an occupant. If an occupant is detected, then it further tracks the occupant via difference of frames and estimation of the direction of motion. The direction of motion is sent to the back-end which resolves the potential adjoining HVAC areas that can be subsequently occupied. In this example, an occupant moving from B towards A will allow the back-end to send an "alert" to the sensor node at A. This sensor

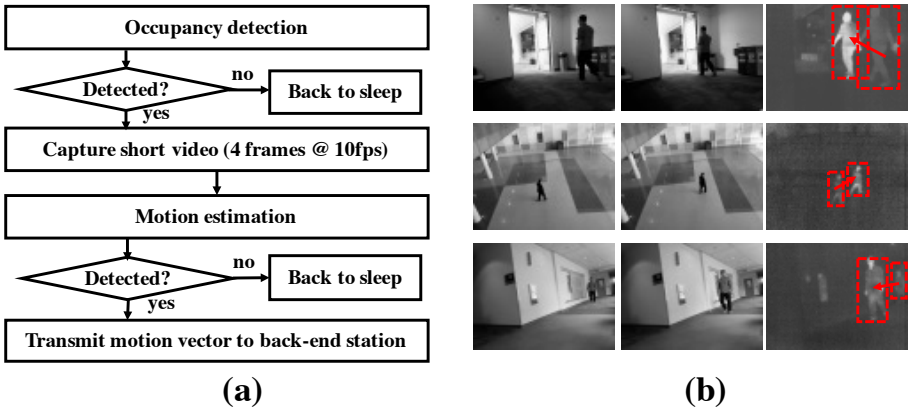


Figure 2.12: (a) Flow chart of occupancy/motion detection and sampling rate; (b) Demonstration of a Case Study.

node, now, increases its sampling rate to reduce the latency of detection. The effective sampling interval T_{eff} is reduced as is shown in.2.11.(b).

For region A, the occupants are either from outside with no sensor node, or from inside the building, that is monitored by the wireless sensor network. The average effective sampling interval $\overline{T_{eff}}$ of this region is:

$$\overline{T_{eff}} = \gamma T_{eff} + (1 - \gamma)T; \tag{2.4}$$

where γ represents the ratio of occupants entering A from adjoining rooms to the total number of occupants entering A. This is illustrated in Fig.2.11.(c). The average sample interval $\overline{T}$ of the sensor node is :

$$\overline{T} = \beta\overline{T_{eff}} + (1 - \beta)T; \tag{2.5}$$

where β represents the average occupancy of the region. Although the average sampling rate is temporarily increased, we increase the default sample interval, which enables us to reduce the overall sensor energy. Combining the above two equations, we obtain the normalized detection latency:

$$d = \frac{2 - \gamma}{2 - \gamma\beta} \tag{2.6}$$

Here we assume that T_{eff} is T/2, the detection latency is proportional to $\overline{T_{eff}}$ and the sensor lifetime is proportional to $\overline{T}$. The numerical results are demonstrated in Fig.2.11.(d). Here we observe that the detection performance is improved significantly, especially in cases of high γ (e.g. $\gamma = 0.8$) and high occupancy β (e.g. $\beta = 0.6$). The corresponding algorithm (implemented at the back-end) and a demonstration of the scheme [67, 53] is shown in Fig.2.12.(a)-(b).

2.5 System measurements

To best evaluate the sensor performance with diverse HVAC systems, two sets of occupancy patterns are randomly generated to simulate an office HVAC environment and a residential HVAC environment. We assume an occupant's arrival time and duration of stay in a particular HVAC region, follow a normal distribution. In the residential occupant model, the mean arrival time is assumed to be 19:00 in the evening with a standard deviation of 1 hour and the mean duration of occupancy of 8 hours with a standard deviation of 2 hours. For an office occupancy model, on the other hand, we assume that people arrive in the office at 9:00 in morning and leave at 14:00 in afternoon with a standard deviation of 1 hour, and leaves the region for a break after 40 minutes with a standard deviation of 10 minutes. Occupants' patterns are assumed to be independent and the region is marked as occupied if one or more occupants appear in the region. We assumed that the mean number of occupants is 5 in the residential environment and 40 in an office environment. Fig.2.13 shows the probability density function of the models and the first subplot of Fig.2.14.(a)-(b) shows a generated example of residential and office occupancy patterns from the model.

As mentioned in Fig.2.6.(d), fusion-based detection demonstrates better miss rate/false positive rate trade-off than single sensor platforms. When the upper bound of the miss rate

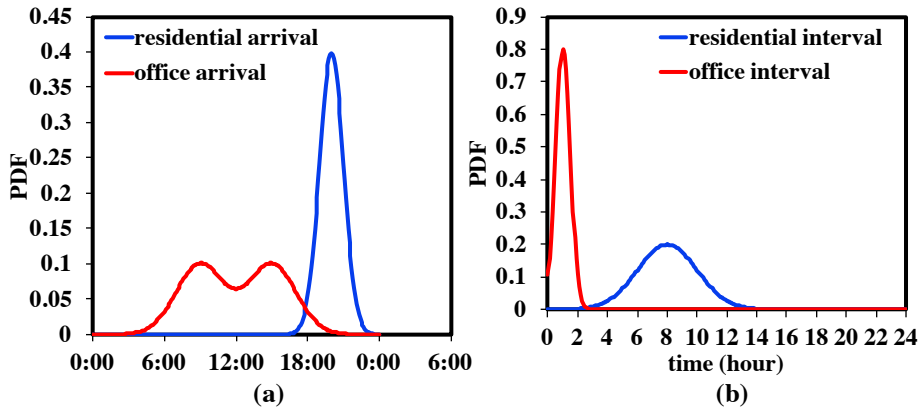


Figure 2.13: (a) PDF of arrival time; (b) PDF of time interval between an individual entering and leaving a region.

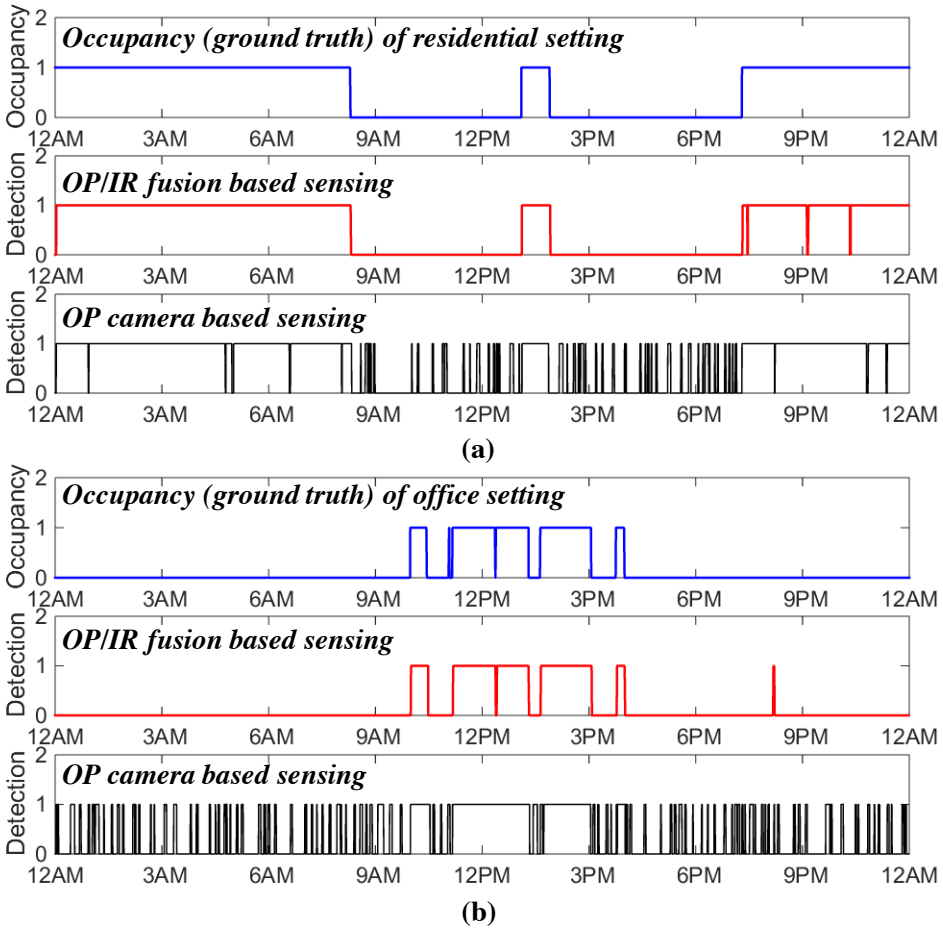
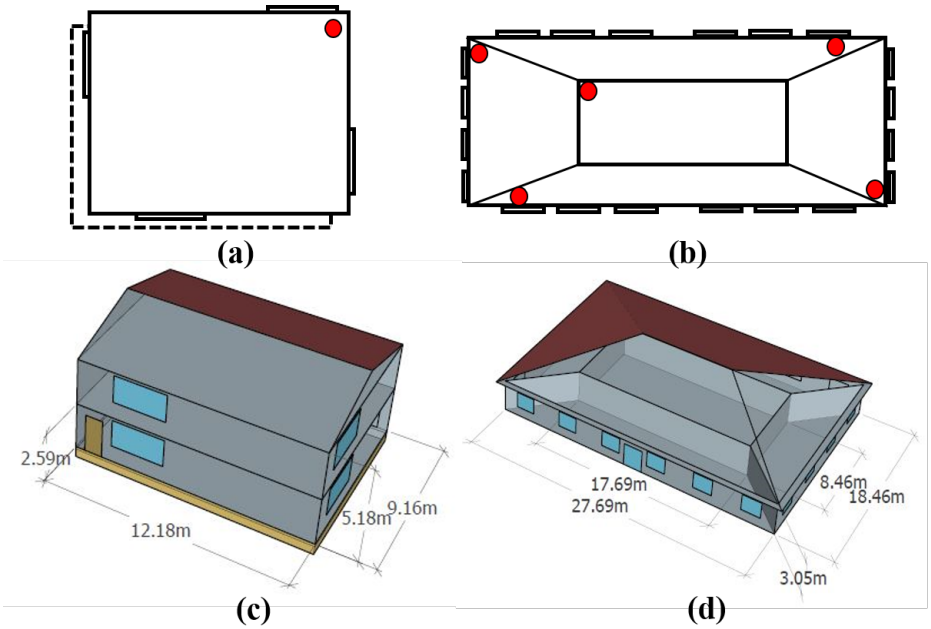


Figure 2.14: Simulated system performance showing occupancy (ground truth), fusion based detection and single sensor based detection in (a) a residential and in (b) an office setting.

is fixed to guarantee a level of human comfort, the proposed fusion-based platform delivers lower false alarm than a single sensor as is shown in the subplots of Fig.2.14.(a)-(b).

To understand the implication of the sensor based system on HVAC energy in typical building scenarios, we use EnergyPlus to model the HVAC energy [45] (parameters shown in figure caption) and the results are discussed for two locations (Chicago and Atlanta). For the results to be representative, we applied reference models from the Department of Energy for both residential and commercial buildings as is shown in Fig.2.15. HVAC control patterns based on the occupancy sensor are generated from Fig.2.14 and the HVAC controlled by programmed schedule is set from 17:00 to 9:00 for the residential scenario



	Total area/m^2	Interior volume/m^3	# HVAC Zone	Insulation (ft^2·°F·h/Btu)		
				Ext Wall	Roof	Floor
Residential	223	578	1	16.4	3.13	1.29
Commercial	511	1559	5	3.45	0.98	2.59

(e)

Figure 2.15: HVAC zone floor plan and sensor placement for (a) residential building (b) and commercial building; corresponding 3D model and dimension in (c) and (d); table in (e) lists all the model parameters.

and from 9:00 to 17:00 for the office scenario. For both sensor-based control and schedule-based control, during summer, the HVAC's hysteretic temperature controls are set to 23C and 26C; during winter these are set to 23C and 20C. More advanced control topologies for the HVAC can further reduce HVAC power as described in [68, 69, 70, 71]. However, the contribution of this work is the data-fusion in-sensor algorithm and advanced control topologies for control will be left for future work.

Simulation in Fig.2.16 show a maximum of 26% (CHI_2) in summer and an average of 18.1-21.4% energy savings are achieved in the fusion-based sensing compared with a schedule-based HVAC control. With our current model, HVAC system saves around 30kWh per day in summer and 55kWh per day in winter on average. At the same time, a single sensor based platform, due to its high false positive rate, consumes more energy than

the schedule-based HVAC control.

In Fig.2.17, the indoor temperature change in one day is demonstrated with residential occupancy pattern shown in subplot1 of Fig.2.14.(a). In both summer and winter, occupancy-based HVAC control outperforms schedule-based control in sampling "unusual" human arrivals, as is shown in highlighted region where resident unexpectedly (1) came back home at noon and stays for a while and (2) arrives home later than usual. In case (1), HVAC is dynamically turned on to provide comfortable environment and in case (2), HVAC is kept off at time of vacancy which saved HVAC energy.

Fig.2.17 shows the trade-off between the HVAC energy savings and detection latency.

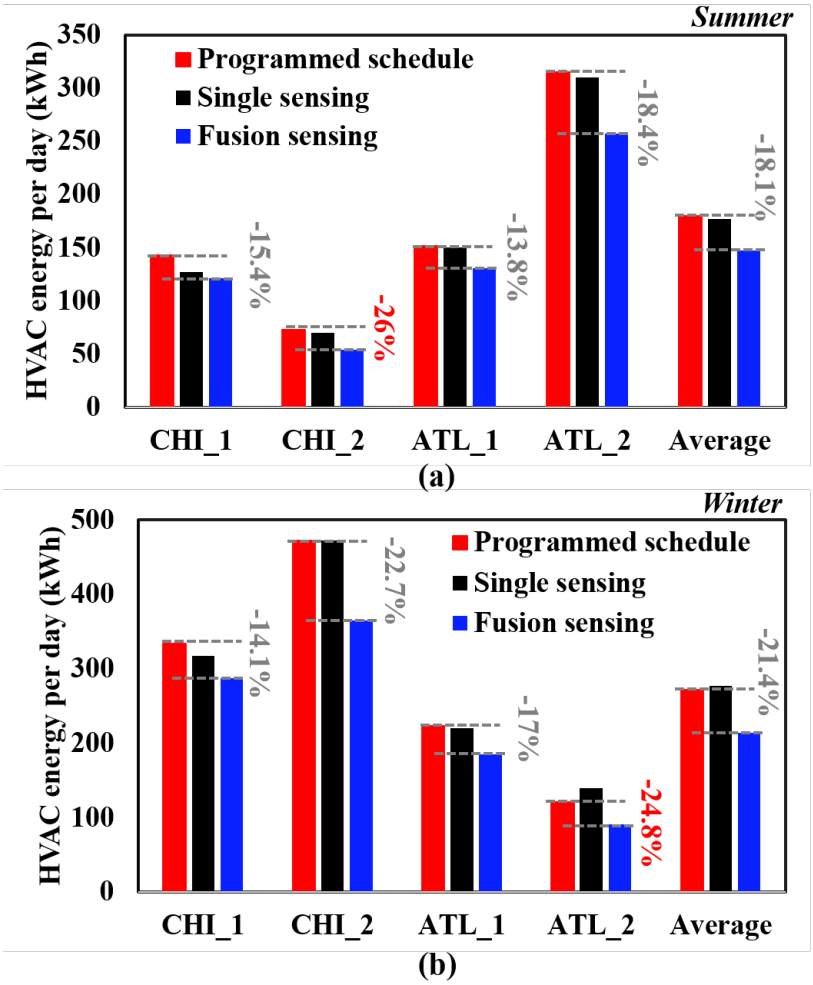


Figure 2.16: HVAC energy consumption per day in (a) summer and (b) winter. "CHI"/"ATL" stands for Chicago/Atlanta and "_1"/"_2" stand for residential/office.

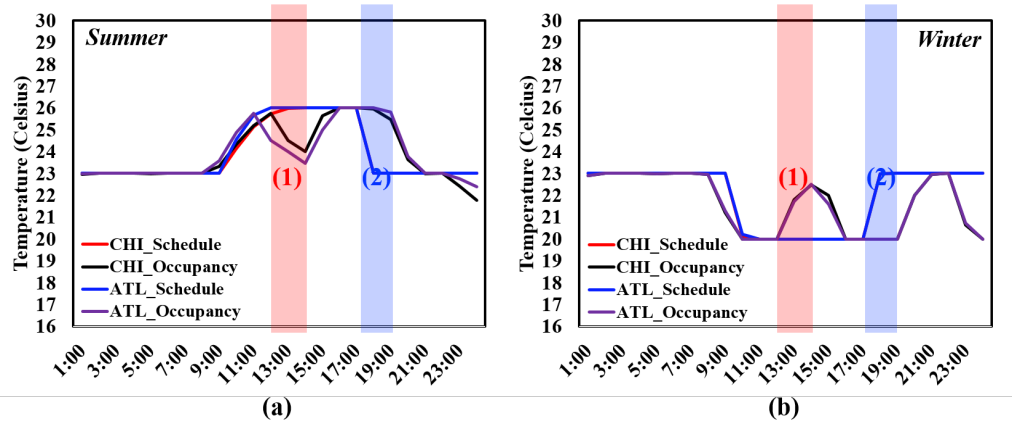


Figure 2.17: HVAC region temperature change in (a) summer and (b) winter.

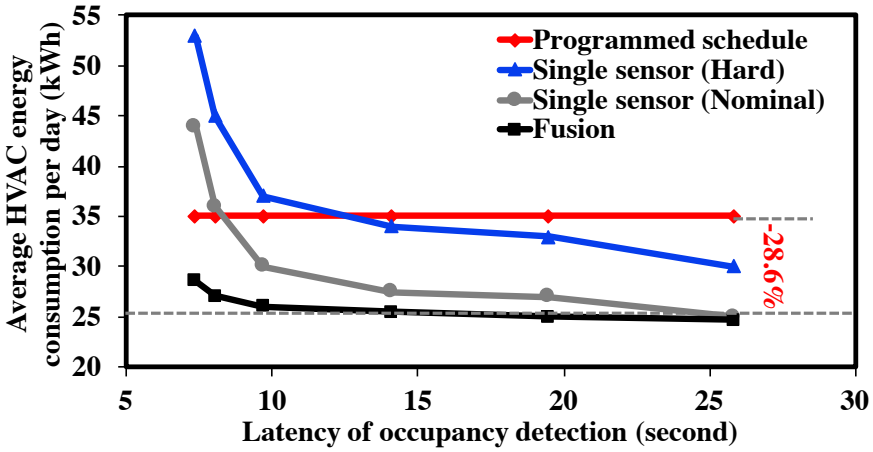


Figure 2.18: HVAC energy vs. latency of occupancy detection.

We observe that energy is saved for all the sensor-based HVAC when more detection latencies are tolerated. We also note that when we have strict detection latency constraints, a single sensor based HVAC control, especially in IR or OP hard cases, performs worse than a simple schedule-based control.

Comparison with state-of-art HVAC occupancy detection platform is demonstrated in the table of Fig.2.19. Depending on the mechanism of detecting occupancy, these works are divided into motion-based approach, such as infrared and camera [48, 49, 72], accessory-based approach, which includes RFID and smart phone [47, 73], as well as the proposed non-intrusive camera based approach. Our proposed work shows high accuracy and significant energy savings from the HVAC system without being intrusive or relying on motion

	System	Sensor	Accuracy (%)	Energy saving (%)
Motion	Y.Agarwal et. al. 2010	Door+PIR	N.A.	10-15
	J. Lu et. al. 2010	Door+PIR	88	28
	Z. Yi et. al. 2010	Camera	80	42
Accessories	N. Li et. al. 2012	RFID	88/62	N.A.
	B. Balaji et. al. 2013	WiFi+phone	86	17.8
Non-intrusive static image	Proposed work	Opical/Infrared Camera	96	13.8-26

Figure 2.19: Comparison with existing literature and competing technologies.

for occupancy detection.

2.6 Conclusion

We proposed a novel collaborative and adaptive template based data fusion algorithm between an OP and an IR camera, which shows significant improvement in miss rate ($5\times$) and false positive rate ($5\times$), extends the lifetime of a wireless sensor lifetime by $3\times$ and achieved a maximum of 26% HVAC energy savings compared to schedule-based control.

CHAPTER 3

COMPUTATION-COMMUNICATION TRADE-OFF IN EI SENSOR NODES FOR WIRELESS VIDEO SURVEILLANCE

In previous chapter, the EI algorithm design is discussed for data-fusion-based occupancy detection. Besides algorithm for data processing, as IoT devices are usually deployed in highly dynamic and complex environments, adaptation of changing environmental conditions is also desirable. Nowadays, IoT devices are applied to diverse environments, and the operating conditions are also constantly changing. As a result, efficient adaptation to the environment is crucial for providing required system level performance. In this chapter, we will discuss an EI control scheme and how the control strategy adapts to time-varying context for improved system-level performance in a wireless video surveillance system. This chapter is a slightly modified version of "Self-optimizing IoT wireless video sensor node with in-situ data analytics and context-driven energy-aware real-time adaptation" published in IEEE Transactions on Circuit and Systems I: regular papers with the dissertation author as the primary author.

3.1 Introduction

With the proliferation of small form factor distributed sensors and Internet of Thing end-nodes, aggregate data transfer to the back-end servers in the cloud is expected to become prohibitively large. For example, 100 image sensors in a sensor network transferring HD data can result in an aggregate throughput of over 1GBps and significantly increase the network's drop rate [62, 63, 64, 65] as is shown in Fig. 3.1. This large amount of data transfer not only results in high energy expenditure and hence low battery life of the sensor node, but it will also results in network congestion producing severe quality of service (QoS) degradation in the form of queueing delay at best, and packet loss or blocking of new

connections in the worst case [74]. This data back-log on cloud servers also precludes any real-time processing and network control, which is a requirement in a myriad of monitoring and sensing applications [75, 76]. Moreover, with the expected rapid growth both in the number of sensors and raw data, the IoT network design itself will only become more complex increasing both the implementation and deployment costs.

To achieve both high energy efficiency in the end-node and seamless network operation, in-situ data analysis capability has to be enabled in the end-node itself [77, 75, 76]. Limited intelligence and decision making, under strict energy constraints, embedded in ubiquitous IoT sensors can reduce the volume of transmitted data by either transmitting only the data of interest or compressing raw data into features or decisions of much smaller volume. It will greatly reduce the volume of data the network has to handle and relieve bandwidth burden on the back-end servers. Although in-situ data-analytics reduces the communication energy at the sensor nodes, it places extra burden on processing. One of the key challenges in IoT nodes is power consumption and system design in pivoted upon reducing the total dissipated power [77]. As we introduce in-situ processing, the computation power increases at the sensor, as it acquires data and analyzes it for possible information content. However, the energy to compute and the energy to communicate are not constants [78, 79,

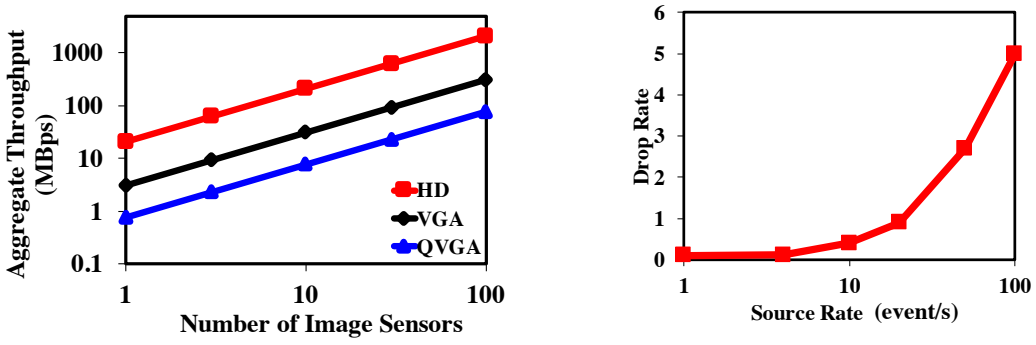
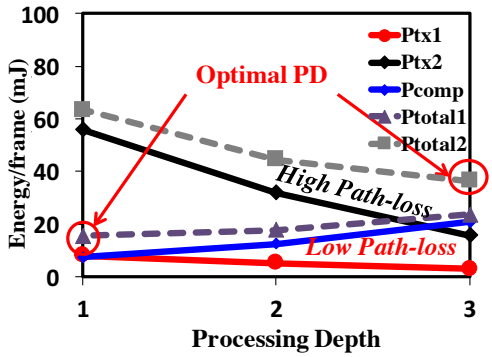


Figure 3.1: (a) Aggregate throughput increases with number of sensor node in the network and the data volume the sensor acquired. (b) Drop rate of the network increased significantly with source rate [64].

Processing Depth (PD)	Operations
1	Object Segmentation+ Compression
2	Object Segmentation+ Compression+ Feature Extraction
3	Object Segmentation+ Compression+ Feature Extraction+ Classification

(a)



(b)

Figure 3.2: (a) Pipelined operations at different processing depth (PD), including temporal difference of consecutive frames (TD), compression (CR), feature extraction (FE) and classification (CL). (b) Power consumption changes with PD and the optimal PD for minimum-power consumption also varies under different channel conditions. For example, a noisy channel results in more embedded processing.

75]. Rather, they are context and environment dependent. For example, a clean wireless channel would lead to lower communication power, with channel adaptive radios. Similarly, if there is no (or little) information contained in the sensed data, then it should be detected early in the processing pipeline. Hence, an energy-optimal system should: (1) allow in-situ data-analytics to extract information from the sensed data to reduce the power overhead of communication, and (2) perform optimal trade-off between the depth of computation and the amount of communication to enable lowest possible power at the sensor node.

This chapter presents a prototypical camera based wireless IoT sensor node for detecting the presence of human beings, with applications in video surveillance. The sensor node supports multiple machine learning algorithms to meet target accuracy requirements. The image processing pipeline (IPP) consists of hardware supported *object segmentation and localization* through temporal difference (TD) followed by *compression* (CR), *feature extraction* (FE) and finally *classification* (CL). We define processing depth (PD) as the stages of computation that are performed in the sensor node, before the data is transmitted to the cloud server. The details of the PD are tabulated in Fig. 3.2a. For example, PD=1 means

that only TD and CR are performed on the sensor node and then the data is transmitted. A PD=2 means that TD, CR and FE are performed before transmission, and so on. The sensor node transmits the output of the processed data and the depth of processing (i.e., PD=1, 2 or 3) for each video frame to the cloud. For PD<3, the rest of the pipeline is implemented in the cloud. An adaptive radio provides power scalable transmission, depending on the signal to noise (SNR) characteristics of the channel. It is intuitive to understand that as the PD increases, the energy cost to compute increases, but the data volume required to transmit decreases, thus reducing the energy cost to communicate. As the channel condition changes (from clean to noisy channel), the minimum energy point also changes. For a clean channel, a lower PD is preferred (as the energy to communicate is low), whereas with increasing path-loss a higher PD is preferred. This is shown qualitatively in Fig. 3.2b, where the energy to compute and communicate (for two channel conditions) have been shown and we note that the minimum energy point is observed at two different PD points. With this motivation, we demonstrate an end-to-end self-optimizing node, which can dynamically adapt the PD depending on the channel condition, to always track the point of minimum total energy. Further, we support multiple CR, FE and CL algorithms depending on the accuracy/power consumption target set by the cloud back-end and the user. Our experimental results show measurements in a dynamic environment where both the information content of the video and the channel conditions are constantly changing. This is due to (1) a mobile sensor node and (2) time varying path-loss.

The complete hardware system consists of an ADI ADSP-BF707 image processor, OV7670 camera sensor and USRP B200 software defined radio. The IPP is implemented on the ADSP-BF707. Measurements have been carried out with a variety of channel conditions and contexts (input image) and, compared with full-transmission and full-computation strategies, we measure a maximum of $4.3\times$ reduction in energy consumption through end-to-end self-optimization. To the best of our knowledge, this is the first work to report fine-grain power management between computation and communication on a self-

optimizing sensor node. We have compared our design with baseline designs where (1) Full-Computation is performed on the sensor node independent of the channel conditions and (2) Full-Transmission of all the acquired data is performed at the sensor node without any "in-sensor" intelligence. The proposed system shows a peak of $4.3\times$ improvement in energy efficiency. We have also compared the design with state-of-art camera based sensor nodes and adaptive wireless systems. These systems do not exhibit any self-optimization between computation and communication. We note $2\times$ to $45\times$ improvement in energy-efficiency (measured in terms of energy/frame) compared to the state-of-the-art designs.

The rest of this chapter is organized as follows. In Section II, the hardware platform is described. Section III introduces the IPP and the embedded human detection algorithm(s) and the tradeoff between detection accuracy and energy-efficiency. The communication system is described in Section IV. Self-optimization between computation and communication in the end-to-end system is discussed in Section IV, followed by experimental results in Section V and finally conclusions are drawn.

3.2 Prototype Hardware Platform

Before we dive into the algorithms and results for in-sensor processing and wireless transmission, let us discuss the hardware platform which forms the basis of the rest of the chapter. In the remainder of the chapter, we will present measurement results to support theory of computation/communication and optimization, based on this embedded platform. The proposed video based sensor platform comprises of camera, image processor, software defined radio, and a PC based controller and configuration control as is shown in Fig. 3.3 and Fig. 3.4. The camera (OV7670) captures 8-bit gray-scale VGA video frames at 10-30fps (frames per second) and consecutive frames, F_i and F_{i-1}, are stored in a 1.53MB off-chip SDRAM. Temporal difference (TD) is computed in the blackfin image processor (ADSP-BF707) with the two subsequent frames fetched from SDRAM to identify, localize and segment a moving object in the image frame. When a moving object is detected, the seg-

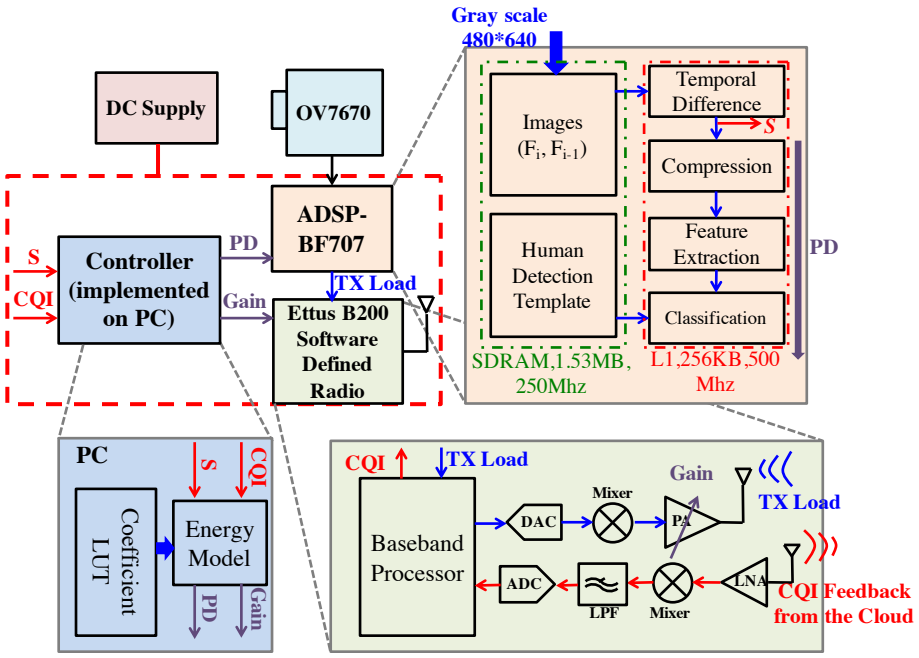


Figure 3.3: End-to-end system architecture showing the different hardware components, the data processing pipeline and the software defined transceiver. CQI is the channel quality index quantified by path-loss and S is the information content size which will be defined in Section 3.3.

mented image of interest is processed through the different IPP stages. Human detection templates are stored in off-chip SDRAM on the board and fetched during CL.

The transceiver (Ettus B200) works in half duplex mode. During transmission, it receives data from the processor (data can be the output of any PD). This data is wrapped in packages with prefix containing information of the algorithm, PD, package length and total data volume. Packages are modulated in GMSK and transmitted at 985Mhz. Channel condition (in terms of path-loss) is evaluated at cloud back-end (which also consists of an identical transceiver board) and sent to the IoT node. The transceiver at the sensor node, adjusts the power amplifier gain accordingly to meet a bit error rate (BER) target, as will be described in Section IV. The configuration settings and end-to-end controller parameters (transmitter gain, PD, choice of algorithm, energy models for each operating condition) are currently implemented in a PC; and can be ported to an embedded hardware for deployment. Platform hardware and architecture is previously discussed in [50].

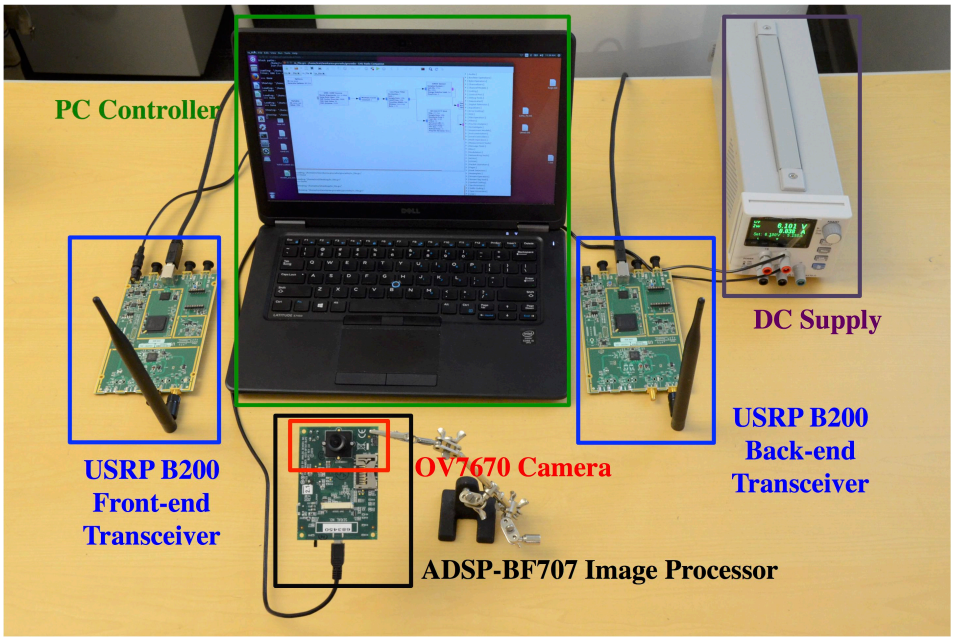


Figure 3.4: Experimental setup showing the system components.

3.3 Embedded Computation

Our current platform is designed for detecting the presence of human beings (henceforth, called human detection) in the field of view. The IPP for human detection is composed of four processing stages: object localization and segmentation through temporal difference (TD), compression (CR), feature extraction (FE) and classification (CL). As discussed in Section I, PD is a direct control knob that allows us to trade-off computation vs. communication at the sensor node. Besides a dynamically tunable PD, the prototype platform offers three algorithm choices with different level of computation complexities and detection accuracy to provide higher level of power-performance trade-off. The target accuracy is set by the cloud back-end and is typically application specific. As is shown in Fig. 3.5, in our design, Algorithm-1 compresses the input frame at the least compression ratio, extracts feature with the most gradients and classifies the feature descriptor with the most computationally-intensive SVM template; and thus achieves best performance in terms of human detection accuracy.On the contrary, Algorithm-3 adopts maximum compression of the acquired frame, extracts the least number of gradient feature and applies the tree based

template, which offers a low-power implementation apt for severely energy-constrained systems. Algorithm-2 is the nominal design point. The design parameters of each algorithm are also listed in the Fig. 3.5. Depending on the trade-off between accuracy requirement and energy budget, a particular algorithm should be selected. This offers programmability on the platform for specific applications and energy constraints. Fig. 3.6 demonstrates how a single frame with a moving object is processed through the IPP and each stage of the IPP are described below.

3.3.1 Objection Localization and Segmentation:

Object localization and segmentation is the pre-processing stage to detect whether a certain frame contains a moving object and segment the object for further computation or transmission. The pre-processing stage prohibits unnecessary computation or communication of following stages when the field of view (FoV) is empty. As pre-processing is always on, the low-power requirement of this algorithm is a primary consideration. There are three major approaches for object activity detection and segmentation: temporal difference [**td1**], model based object localization [80, 81] and optical flow [82]. Optical flow method can

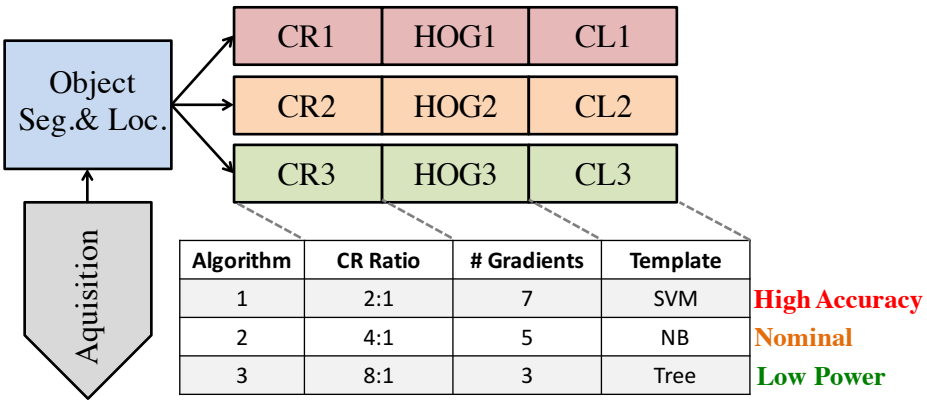


Figure 3.5: Embedded human detection computation and design points of different algorithms/operations.Algorithm-1 (highest accuracy) applies CR ratio of 2:1, 7 feature gradients and SVM classification template; Algorithm-2 (nominal) compresses input frame 4 times, extracts 5 gradients per feature and applies NB human detection template; Algorithm-3 (most energy-efficient) heavily compresses input frame 8 times, extracts 3 feature gradients and classifies with the tree template.

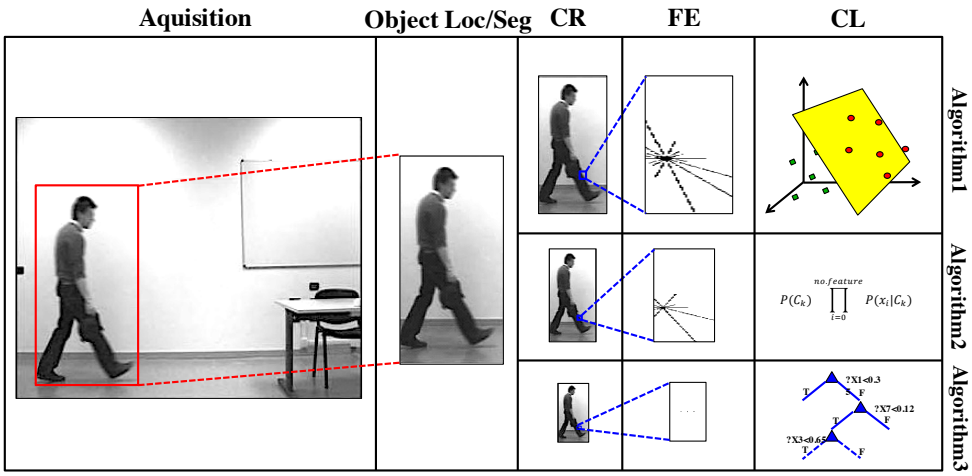


Figure 3.6: Algorithm demonstration with a real video frame.

obtain complete information and detect the moving object from background better, but requires clustering, which is computationally expensive and unsuitable for real-time IoT operation. Model based background subtraction relies heavily on dynamically calibrated background models, which has a large overhead in an embedded systems, especially under strict power constraints. Compared with optical flow and model-based background extraction, temporal-difference computes moving object area with the least operation and consumes least energy.Hence, in the current implementation, we use temporal-difference for its simplicity and high energy efficiency [80] in the low-power pre-processing stage. In the temporal difference method, we subtract two consecutive video frames. The pixels whose difference is greater than a certain energy threshold, E_{th}, are labeled as activated pixels with label value of 1. Otherwise, label value 0 is assigned. This can be summarized as:

$$D_{\mathrm{i}}(m,n) = |F_{\mathrm{i}}(m,n) - F_{\mathrm{i\text{-}1}}(m,n)| \tag{3.1}$$

$$L_{\mathrm{i}}(m,n) = \begin{cases} 0, |D_{\mathrm{i}}(m,n) - D_{\mathrm{i\text{-}1}}(m,n)| \leq E_{th} \\ 1, |D_{\mathrm{i}}(m,n) - D_{\mathrm{i\text{-}1}}(m,n)| > E_{th} \end{cases} \tag{3.2}$$

The area of interest is defined as the pixels within the rectangular boundary with label value of '1'. We quantify the "information content" (S) of a frame as the number of activated pixels (normalized to the total number of pixels) and it forms a consistent measure of context in camera based sensor nodes. If information content is less that 3.125% (60×40 in a QVGA frame), we do not perform any further processing and the entire system is gated till the next frame is captured.

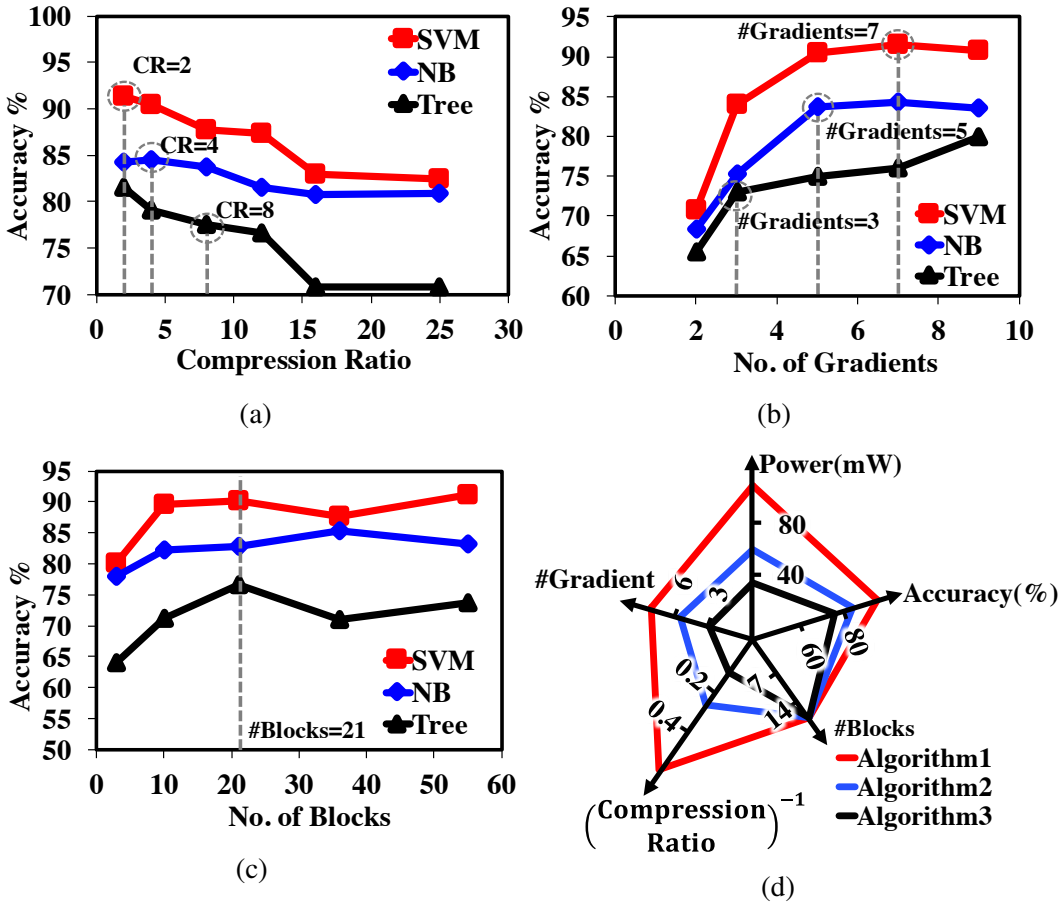
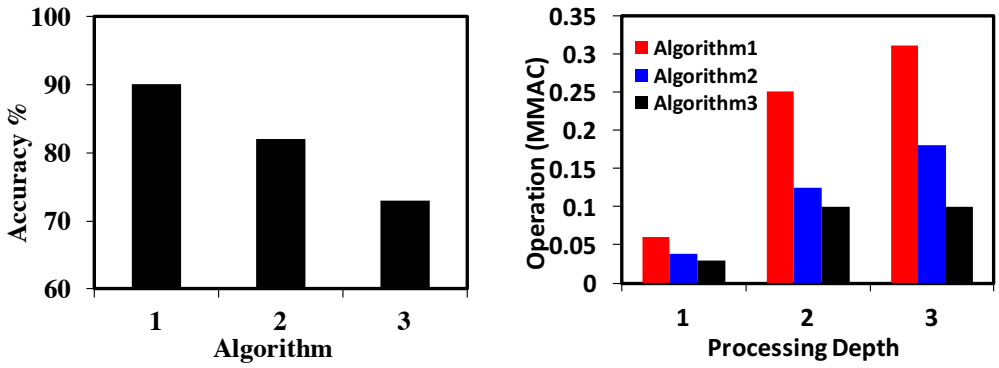


Figure 3.7: (a) Measured detection accuracy vs. compression ratio. (b) Measured detection accuracy vs. number of gradients extracted from HOG feature extraction. (c) Measured detection accuracy vs. number of blocks to extract feature vectors in HOG feature extraction. (d) Power consumption and accuracy at design points in different algorithms.



(a) (b)

Figure 3.8: (a) Measured human detection accuracy with three different algorithms. (b) Number of estimated operations in millions of multiplication-accumulation-counts (MMAC) for different algorithms/depths.

3.3.2 Compression

The second stage of IPP is image compression. The purpose of compression is to reduce the amount of data to compute or communicate while maintaining a target accuracy requirement. This is simply performed by averaging the pixel values over a sliding window. In our design, compression further scales down the segmented image from pre-processing by evenly averaging pixels at certain compression ratio. CR1, CR2 and CR3 represents increasing compression as shown in Fig. 3.5.

3.3.3 Feature Extraction

Feature extraction derives informative and non-redundant values to facilitate the subsequent stages to generate better classification results. In human detection, feature extraction is crucial to discriminate human from cluttered background. Different feature descriptors are available, including wavelets, SIFT and HOG. Among all feature extractors, Histogram of Gradient (HOG) is chosen for its excellent performance and large INRIA human dataset availability [56, 57]. HOG first divides the input image matrix evenly into M$\times$N cells. Gradient angle and gradient magnitude of each pixel are computed. Each pixel within the cell votes for an orientation-based histogram channel by comparing gradient angle with angle

bins with weight of gradient magnitude. Angle bins evenly spread on $(-\pi, \pi]$ range and number of bins is N_{bin}. Then the spatially connected cells form a block of size (M-1)×(N-1) to be locally normalized to account for changes in illumination and contrast where M and N stands for number of rows and columns of cells. The hardware supports three FE options, as shown in Fig. 3.5.

3.3.4 Classification

Classification is the final step in the IPP. The classifier is trained offline in testing phase and classification template is generated and stored in the SDRAM. Different machine learning classifiers have different performance-power trade-offs. We employ three different classification schemes depending on the target accuracy set by the cloud back-end depending on the application. Based on our simulations, we support Support Vector Machine (SVM) for highest performance, Naïve Bayes classifier (NB) for nominal performance, binary tree classifier for highest energy efficiency, as three classifiers to offer different trade-offs of complexity/accuracy in human detection.

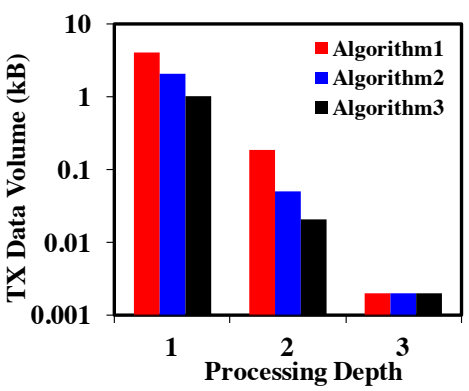
Naïve Bayes (NB) classifier [83, 84, 85] assumes strong independence between individual descriptors and applies Bayes' theorem, which describes stochastic event based on related conditions, on test data to predict class:

$$P(C_{\mathrm{k}}|\vec{x}) = \frac{P(C_{\mathrm{k}})}{P(\vec{x})} \prod_{i=1}^{N_{\mathrm{in}}} p(x_{\mathrm{i}}|C_{\mathrm{k}}); \tag{3.3}$$

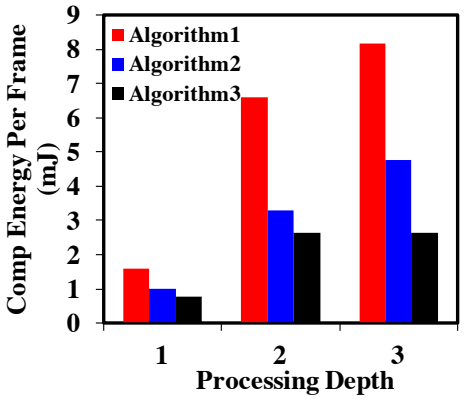
$$C_{\mathrm{K}} = argmax_{\ \mathrm{C_k}}\ P(C_{\mathrm{k}}|\vec{x}) \tag{3.4}$$

where C_k is the kth class and $\vec{x}$ is the input test descriptor. $P(C_k)$, $P(\vec{x})$ and $p(x_i|C_k)$ are constants and obtained from training and the predicted class, C_K, is the one with highest conditional probability for all classes. In the current set-up, there are only two labels: human and non-human.

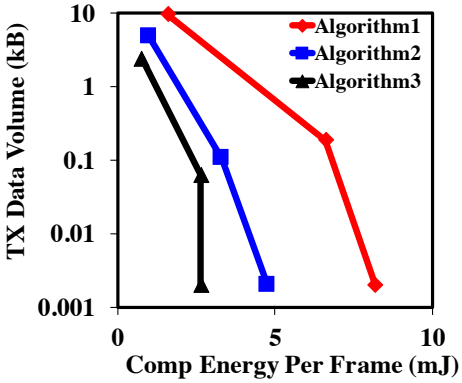
Binary tree or decision tree, is composed of nodes, branches and leaves representing test, test-outcome and labels respectively [86, 87]. The configuration of a tree, including shape and test condition on each node is obtained by observation. The tree classifier in our application is binary, which means, starting from root node, on each node, one of the predictors in feature set is compared with a certain threshold. Then it either goes left or right depending on comparison result till it reaches the one of the leaves which contains one of the classification results.



(a)



(b)

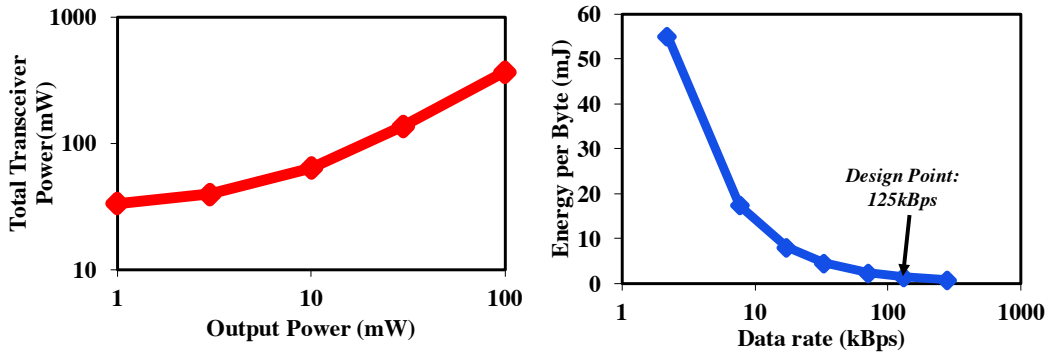


(c)

Accuracy Requirement	Algorithm	Estimated E_{comp} per Frame (mJ)			Estimated TX Data Volume (kB)		
		PD=1	PD=2	PD=3	PD=1	PD=2	PD=3
Highest	1	1.6	6.6	8.2	4	0.15	0.002
Nominal	2	1	3.3	4.8	2	0.11	0.002
Lowest	3	0.8	2.6	2.61	1	0.06	0.002

(d)

Figure 3.9: (a) Measured transmission load vs. processing depth with different algorithms and PD. (b) Measured front-end computation energy per frame vs. processing depth. (c) Estimated Tradeoff between transmission data volume with computation energy (d) different detection accuracy requirements result in different algorithm chosen, computation energy (Ecomp) and transmission data volume



(a) (b)

Figure 3.10: Measured (a) transceiver power vs. output power. (b) energy per byte vs. data rate.

3.3.5 Comparative Analysis of Classification Schemes

The three classification schemes have been mapped to the ADI camera processor and optimized for minimum area on the on-board memory. Benchmarking is carried out on the INRIA human dataset [56, 57]. Tree classifiers use cascaded comparators of depth 10, and are the most energy efficient scheme. SVMs demonstrate highest performance but require more than 500 support vectors and hence dissipate the highest power. NB shows nominal performance and power dissipation. Fig. 3.7 illustrates how detection accuracy changes with CR, number of gradients and number of blocks in different classification algorithms. Accuracy improves when CR is low and more gradient features where number of blocks in feature extraction does not show strong tendency. The design parameters for each algorithm selected for our platform are also denoted in the figure. The compression ratio are designed as 2:1, 4:1 and 8:1 for three algorithm with 3, 5 and 7 gradient orientations in each block of feature extraction. Number of blocks in extracting gradient orientation is designed to be 21 for all algorithms.

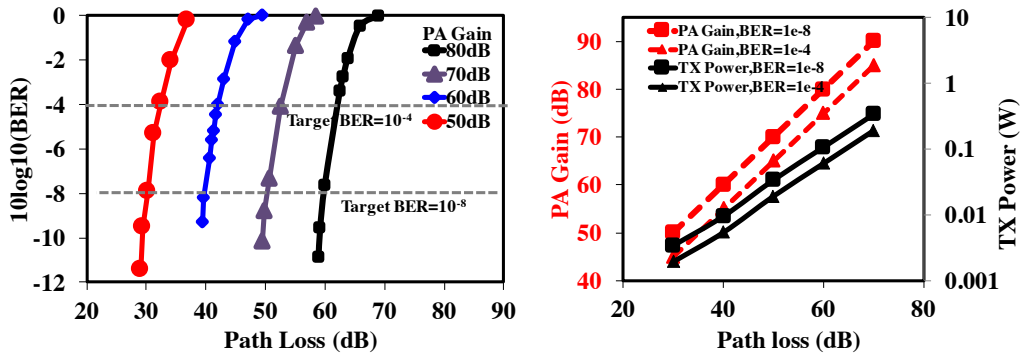
Accuracy measurements in Fig. 3.8a were carried out on human detection database, INRIA [56] because of its relevance to surveillance. The design parameters are chosen from Fig. 3.7. Algorithm-1 is designed to provide a target accuracy of 91% while Algorithms-2 and 3 provide target accuracy of 83% and 77% respectively. Fig.3.8b illustrates how the

number of computations (in terms of 10^6 MAC operations) changes with both the algorithm of choice and the PD. Higher accuracy and deep embedded processing suffer from heavy computation which is expected to result in high computation energy expenditure.

As the PD is increased, the amount of data required to transmit to the backend (including all the header information) is reduced. Fig. 3.9a illustrates the transmitted (Tx) load (i.e., the amount of data to be transmitted per frame) for each computation depth. Fig. 3.9b, illustrates the measured computation energy per frame for the three different algorithms and PDs as discussed above. We note that the lowest computational energy of 0.71mJ/frame is recorded for Algorithm-3 and PD-1 while the highest computational energy of 8.2mJ/frame is measured for Algorithm-1 and PD-3, thus showing a span of 8X/9X depending on the choice of algorithm and PD. We also note that as the computation energy at the sensor node increases (higher PD), the total data volume decreases sharply thus allowing a smooth trade-off in the cost of computation and communication. Key results are tabulated in Fig. 3.9d.

3.4 Adaptive Wireless Communication

Wireless communication conventionally is the major cause of energy expenditure and shortened lifetime of wireless sensors, especially when the sensors are experiencing expanding



(a) (b)

Figure 3.11: Measured (a) Bit-error-rate vs. path-loss under different PA gain. (b) PA gain and transceiver power vs. path loss under BER requirement of 10^{-4} and 10^{-8}.

bandwidth, rapid growth of nodes and ever-increasing data volume with the development Internet of Things [75, 76]. To implement energy-efficient wireless design on SDR (software defined radio), the power/energy characteristics of the adaptive radio is first explored. As is shown in Fig. 3.10a, transceiver power, first dominated by standby power at low loads, increases with output power and dynamic power gradually dominates which is generally the case with noisy channels or long-distance transmissions. In Fig. 3.10b, it is observed that with the increase of data rate, energy per byte transmitted decreases tremendously. In our system, data rate is set at 125kBps by GNUradio.

Traditionally, transceivers are designed for the worst-case, hence maximum power consumption, to guarantee target performance, such as bit-error-rate (BER). However, as channel condition of wireless sensors varies significantly from time to time [**ber**], adaptive wireless communication is desired which adjusts the transceivers dynamically to operate marginally with respect to performance according to temporal channel quality to save energy [88, 78, 79, 89, 90, 91, 92, 93, 94, 95]. Channel quality is affected by (1) Path Loss (2) Interference Strength. (1) can be compensated by increasing transmitted power amplifier (PA) output power, (2) can be handled by increasing receiver linearity. Since we focus on co-optimizing computation and transmitter power, we mostly focus on (1) in this work. .

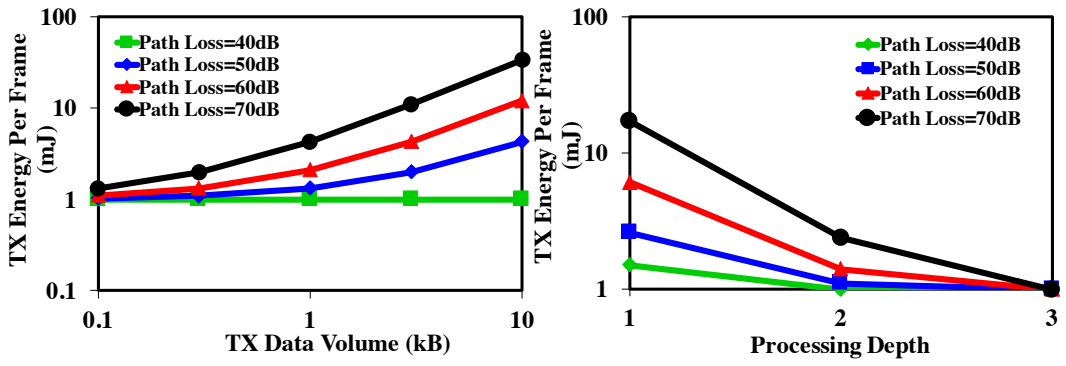


Figure 3.12: Measured (a) transmission energy per frame vs. transmission data volume under various channel conditions. (b) Transmission energy per frame vs. processing depth under different path-loss conditions.

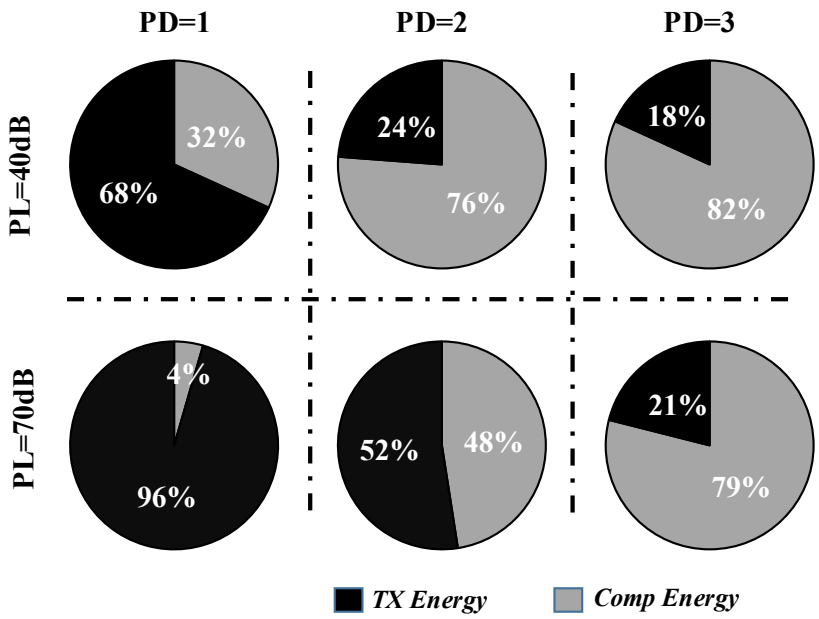


Figure 3.13: Breakdown of computation energy and TX energy in different processing depth and path-loss.

Path-loss in dB is expressed as [96]

$$Path\_loss = 20log_{10}(\frac{4\pi df}{c}) \tag{3.5}$$

Here d is distance, f is the carrier frequency and c is the speed of light. In our design, the carrier frequency is 985Mhz.

To compensate for path-loss, the power amplifier gain is adjusted dynamically to guarantee minimum BER. Measured BER vs. path-loss for different PA gains of the SDR are shown in Fig. 3.11a. The PA gain and the total transmission power required to meet a target BER=10^{-8} and BER=10^{-4} for different path-loss are also shown in Fig. 3.11b. For the rest of the chapter, we will use these two target BERs. BER=10^{-8} is a conservative target ,which represents minimal error detection/correction and channel coding and high communication energy. On the other hand, a more relaxed BER target of 10^{-4}, with complex channel coding employed, illustrates usage models where the energy cost of computation can dominate the energy cost of communication, particularly for cleaner wireless channels.

In this chapter, we have not considered the network aspect of the wireless node. Hence, we present results for both a conservative BER target and a relaxed BER target that encompasses typical ranges for wireless nodes. We measure the total transmitted energy as a function of the total transmitted data volume (also referred to here as Tx load). For low path loss, the standby power dominates, however with increasing path loss and PA gain we see a near-linear increase in total transmission energy as a function of the data volume (Fig. 3.12a). Since, the volume of transmitted data decreases with PD, we can now estimate the total transmission energy per frame of video data as a function of PD, as shown in Fig. 3.12b. With clean channel (40dB path-loss), transmission energy per frame is 1mJ for transmission after PD1, while for noisy channel (70dB path-loss), transmission energy per frame can be as high as 17mJ.

The energy breakdown of the system is demonstrated in Fig. 3.13. Here, we can observe that in a noisy channel with a path-loss=70dB, transmitter energy occupies more total budget as compared to a clean channel. At the same time, with deeper processing depth, transmitter energy can be saved at the expense of computation energy. The overall self-optimization of total energy will be introduced in the section V.

3.5 Self-optimization Procedure and System Setup

In the previous sections we have seen the strong trade-off between transmission energy, PD and the algorithm of choice. A self-optimizing system needs to be cognizant of this, and adjust its operating point dynamically based on the choice of algorithm and channel conditions.

3.5.1 Energy Model

We first develop a model for the total energy of the sensor node. The total energy, E, includes computation energy, E_p, and communication energy, E_{TX}; and is a function of temporal variables of information content (S), processing depth (PD), and path-loss, (PL),

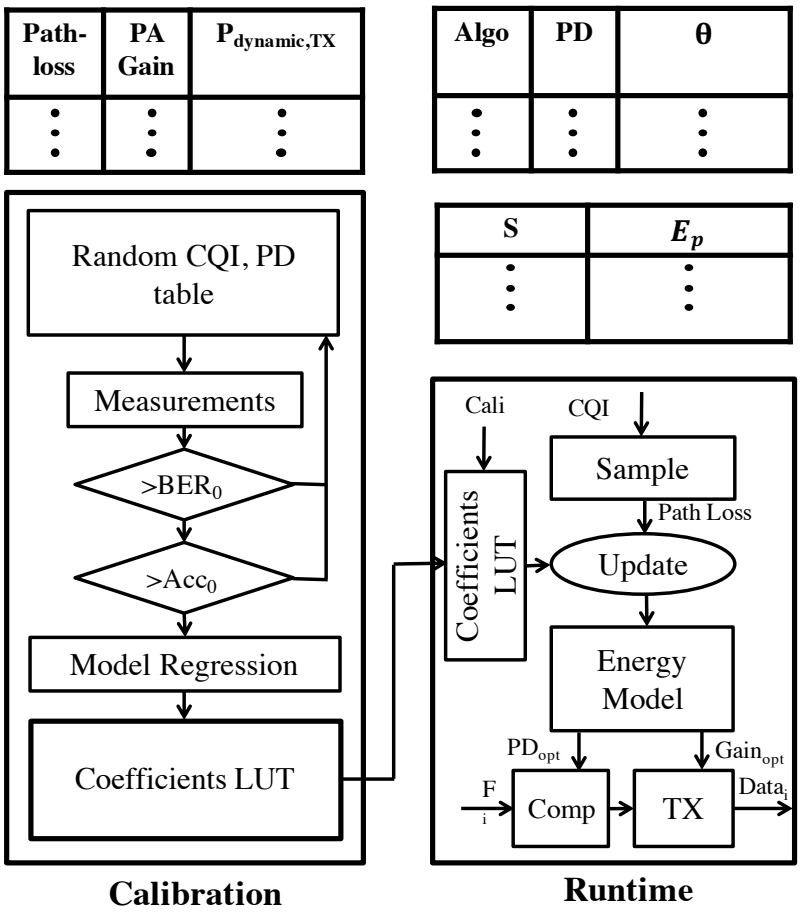


Figure 3.14: Calibration and runtime self-optimization scheme.

under the constraint of accuracy requirement, (Acc_0), as defined by application/cloud server when choosing the most-energy efficient algorithm, ALG.

$$E = E_{\mathrm{p}} + E_{\mathrm{TX}} = f(S, PD, PL), Acc(ALG) > Acc_0; \tag{3.6}$$

Once the most energy-efficient algorithm is chosen according to minimum accuracy requirement, computation energy is only a function of information content and processing depth independent of path-loss and it can be further decomposed into dynamic energy and static energy per frame. With processing period fixed at T, i.e., 1/frames per second, E_p changes with processing time (τ_p), a function of information content and processing depth. Large information content size, deep embedded processing and more complex algorithms

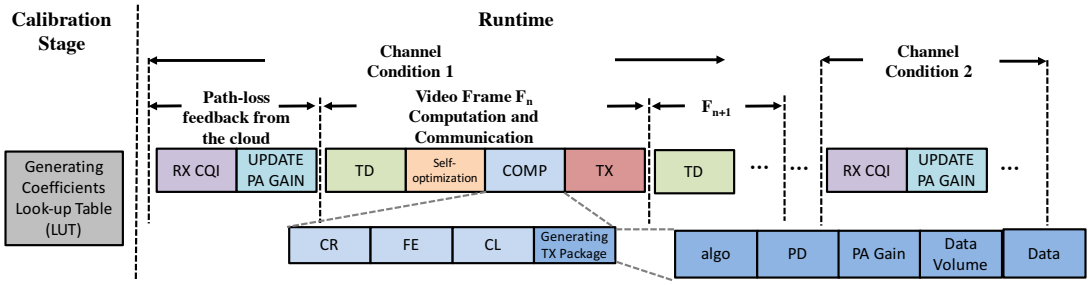


Figure 3.15: Data packet configuration and modes of transmission-reception for the wireless link.

will result in high computation energy. $P_{\text{dynamic,p}}$ and $P_{\text{static,p}}$ are the dynamic processing power and static processing power respectively which are obtained from the image processor measurement. The processing energy can then be expressed in terms of S, PD and other parameters as

$$E_{\text{p}} = f_1(S, PD) = P_{\text{dynamic,p}} \cdot \tau_{\text{p}}(S, PD) + P_{\text{static,p}} \cdot T = \theta_{\text{ALG,PD}} S + E_{\text{static,p}} \tag{3.7}$$

where $\theta_{\text{ALG,PD}}$ is model coefficients of algorithm ALG at processing depth PD which is fitted via regression during pre-deployment testing and calibration

Communication energy is modeled as a function of PL, power amplifier (PA) gain and the static power. The total energy to transmit each video frame is modeled as

$$\begin{aligned} E_{\text{TX}} = f_2(S, PD, PL) &= P_{\text{dynamic,TX}} \cdot \tau_{\text{p}}(S, PD) + P_{\text{static,TX}} \cdot T \\ &= P_{\text{dynamic,TX}}(PL) \cdot \frac{\Gamma_{\text{ALG}}(S, PD)}{DR} + E_{\text{static,TX}} \end{aligned} \tag{3.8}$$

where $\Gamma_{\text{ALG}}(S, PD)$ is transmission load when processed by algorithm-ALG, processing depth of PD and information content of S, and DR is the data rate.

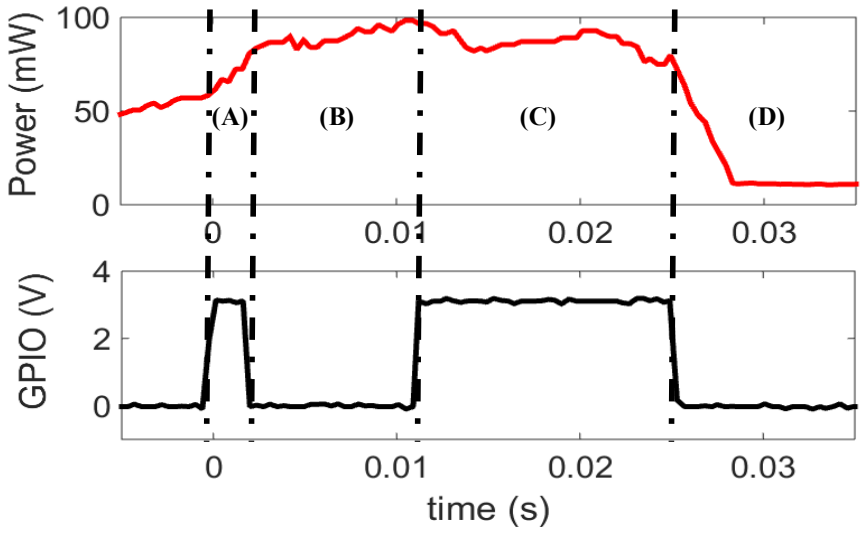


Figure 3.16: Measured embedded computation power consumption where transient congrol signal of each processing stage is indicated by GPIO output voltage level: (A), object segmentation and localization through temporal difference (TD) together with compression (CR); (B), feature extraction(FE); (C), classification (CL) and finally (D), idle power down state. Note that alternative opertions have alternative active-high and active-low control signals. For example, (A) is active-high, (B) active-low and so on.

3.5.2 Self-optimization Procedure

The over-all system first characterizes itself before deployment. On the test-bench, for different algorithms, PD and path loss conditions, the system performs energy calibration and determines the total energy for each IPP task and transmission. Then the system populates a look-up table (LUT) which contains information about possible operating conditions. This is currently implemented on a PC, but can be embedded if required. This calibration step can use external or embedded sensors (power/current sensors); and, in the present system we perform the calibration using external on-board sensors.

Calibration of the system is performed during test phase. This procedure is illustrated in the flow-chart shown in Fig. 3.14. The key algorithmic steps before the IoT node is deployed are:

1. The algorithms (combination of different compression ratios, feature extraction methods and classifiers) are characterized on a known (INRIA) data-base during design.

The accuracy of the algorithms for the task at hand are determined.

2. During calibration phase, models for energy dissipation are constructed. A random value of path-loss is generated. A corresponding minimum power amplifier gain that satisfy the target BER is measured and the gain together with its $P_{dynamic,TX}$ are stored in the corresponding LUT entries.

3. LUT entries for the coefficient θ are populated for each algorithm and processing depth. Assuming a linear relationship and to avoid over-fitting, ten processing energy measurements (E_p) against ten random information sizes (S) from a test video per PD and algorithm are used in the current setup. We use regression to calculate θ. Videos in this calibration stage are obtained from ViSOR data-set, "Outdoor, Unimore D.I.I setup" category. It encompasses a large range of information content, from pixel sizes of 2400 (60×40) to 21600 (180×120). This allows us to obtain a comprehensive and accurate energy model which is critical for the success of the design. During run-time we test the setup with a real-time system with hours of videos obtained from the OV7670 image sensor. This allows us to obtain accurate measurements of energy consumption during operation and perform online optimization between computation and communication energy. It should be noted that to train the system for human detection we used the INRIA image data-set, as has been mentioned, and performance/accuracy testing was done on hours of real-time videos acquired with the final system setup.

After deployment, information about path-loss is sent from back-end cloud to the front-end platform periodically (every 1s) and the minimum power amplifier gain needed to overcome path-loss is updated. Then the energy model estimates the energy for all the IPP blocks with respect to the information content. Then the system chooses the PD for minimum energy of operation. The PD information, algorithm, transmission gain and energy for IPP blocks are packed into the frame header and transmitted. This is used by cloud

server for back-end processing. The calibration and run-time self-optimization scheme are shown in Fig. 3.14 and data/operations in time domain is shown in Fig. 3.15.

Upon obtaining accurate coefficients, the overhead of the self-optimized system is limited to storing the model parameters and modeling the computation/communication energy. The model, including PL-PA gain table, will consume no more than 40 bytes of memory in double-precision. For the system running at 10 frames per second, the maximum computation needed for the energy estimation is 70 MAC/second. For the overall system, both the model storage and energy estimation overheads are negligibly small.

3.6 End-to-end System Demonstration and Measurements

The algorithms are implemented on ADI-BF707 image processing board and computation power consumption is measured. An example of measured power and the processing steps is shown in Fig. 3.16. We can observe the different processing steps through GPIO output (the IPP steps are alternatively active high and active low), and the corresponding power consumption. During pre-deployment calibration, the LUT is populated and the energy models are constructed for varying path-loss and information content of the captured video frames. Based on the LUT data, the system chooses the operating mode for minimum energy per frame. This is shown in Fig. 3.17 where different PL scenarios are examined. As the PL increases, the self-optimizing sensor node always chooses the most power optimal PD. We note that the increasing path-loss will result in more embedded computation and total energy is saved on the self-optimizing platform. Also, improved energy-efficiency will be achieved with low-power algorithm, Algorithm-3 for example, or lower target BER, i.e. 10^{-4}. Comparisons on total energy per frame is also demonstrated among different design strategies in Fig. 3.18. We compare the results of the proposed system vis-a-vis two static designs. These are:

1. Full-Transmission: In this design the sensor node only performs image acquisition, localization and compression, and then transmits the entire video data.

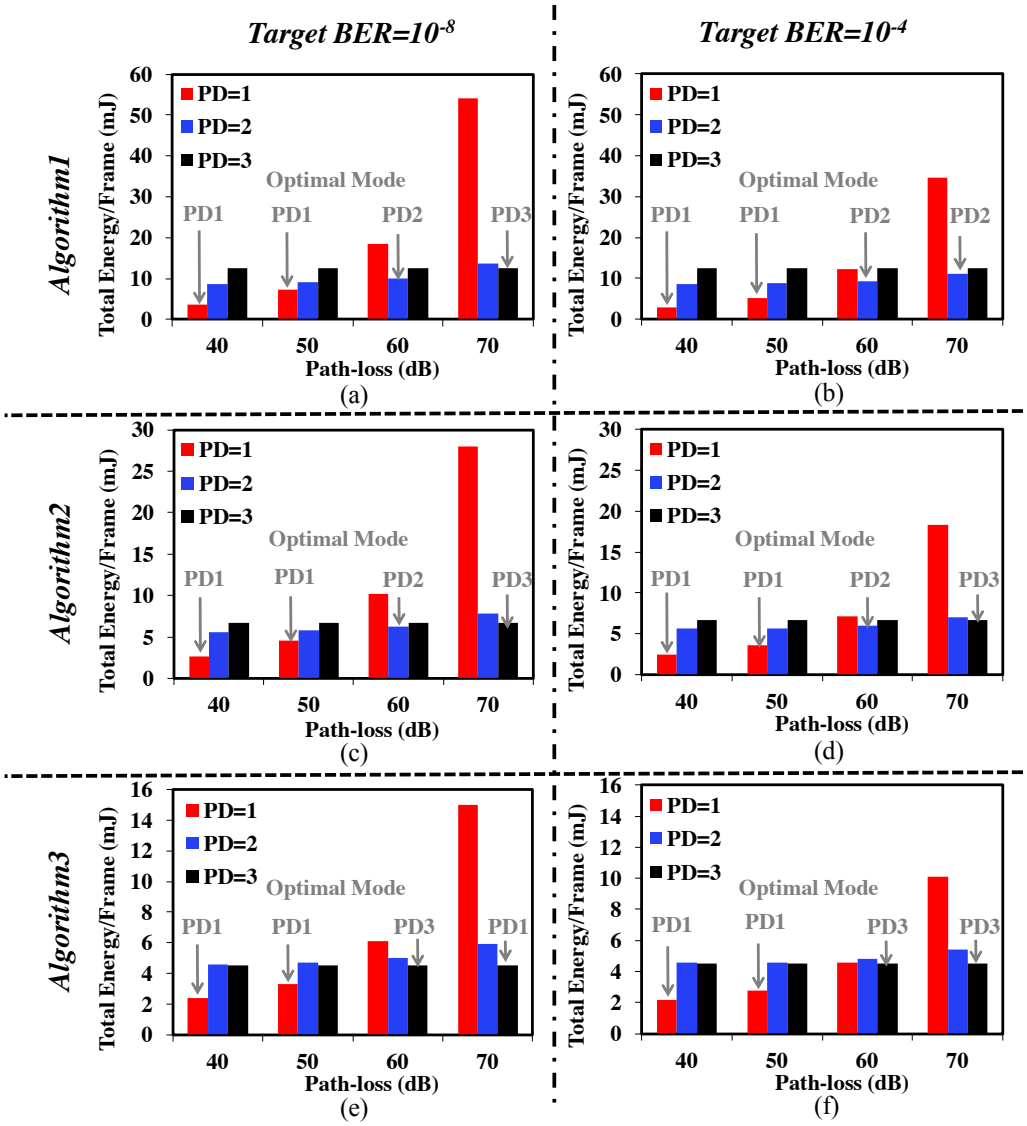


Figure 3.17: Measured total energy (computation+communication) per frame for different PD with increasing path-loss. Experimental results are demonstrated for the three algorithms described here and two BER targets.When path-loss is high, the general trend is that optimal mode moves to more front-end embedded processing.

2. Full-Computation: In this design the sensor node performs all the tasks in the IPP without considering the energy cost of computation, independent of the channel conditions.

We note that by properly balancing the energy for computation and communication, the proposed system always operates at minimum energy point. We measure peak sav-

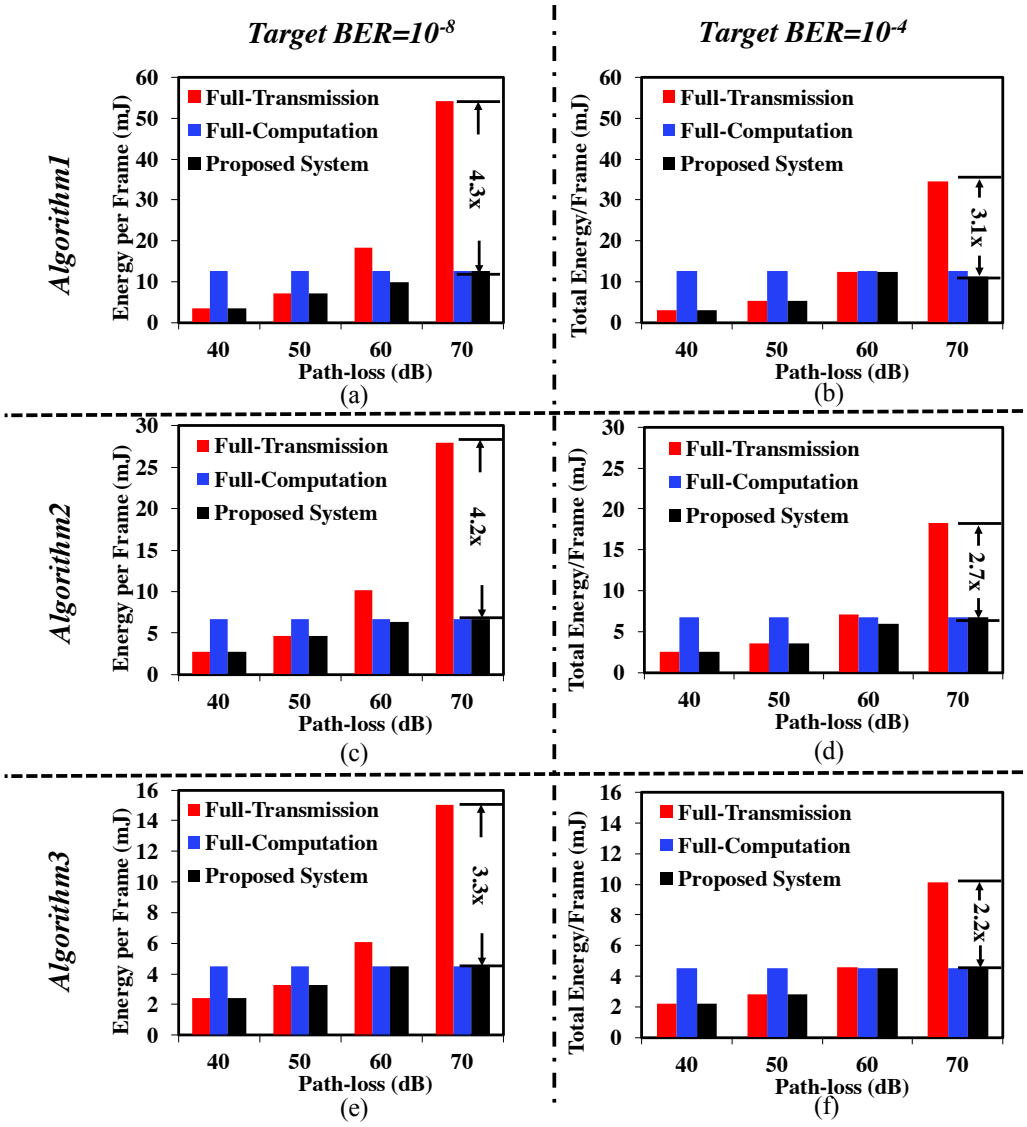


Figure 3.18: Measured total energy (computation+communication) per frame for the proposed system comapred against two static designs.Experimental results are demonstrated for three algorithms and two BER targets.

ing of $4.3\times$ at 70dB path-loss, operating with Algorithm-1 and target BER of 10^{-8}, when compared with baseline design (Full-Transmission Design). For a target BER of 10^{-4}, the proposed system shows $2.2\times$ to $3.1\times$ peak savings. A random path-loss scenario is generated and its impacts on PD, PA gain, computation energy per frame, communication energy per frame and total energy is demonstrated in Fig. 3.19.(a). We note how in transient mode the system operated at the correct PD to track minimum overall energy by trading compu-

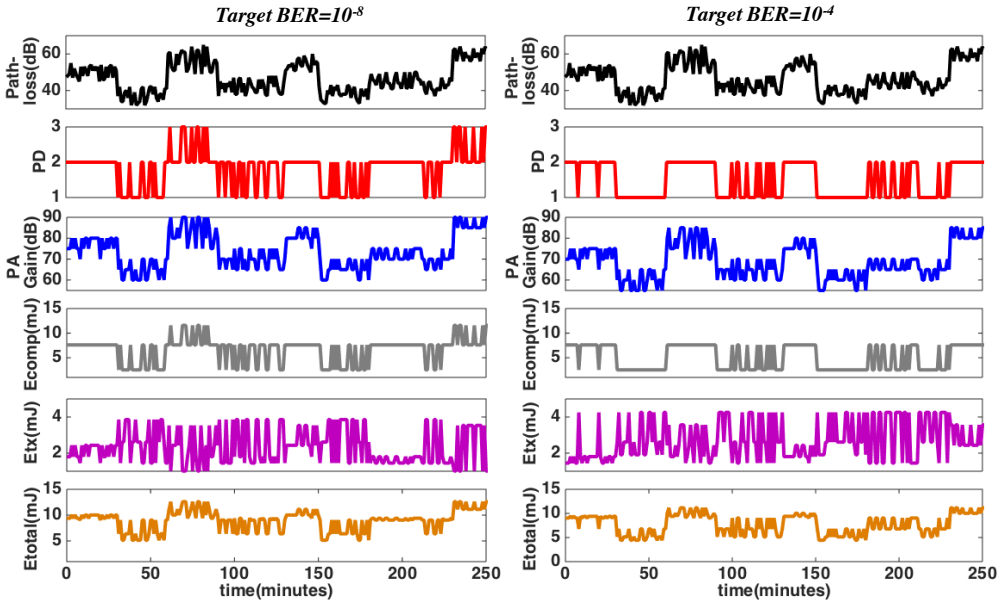


Figure 3.19: Case Study: Random and dynamic path-loss condition created by a mobile IoT node and the corresponding PD, PA gain, computation, transmission and total energy per frame under BER constraints of (a) 10^{-8} and (b) 10^{-4}.

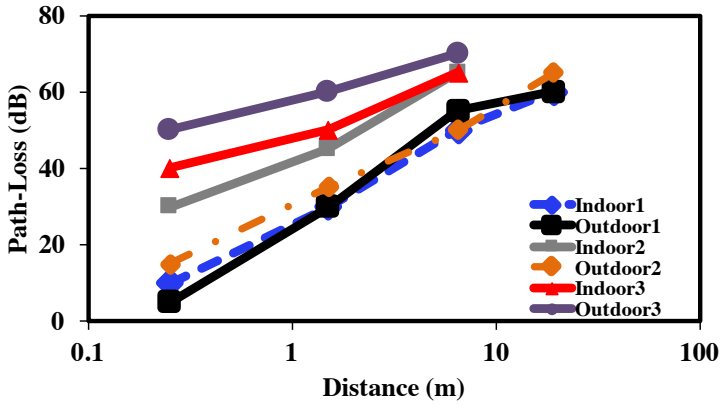


Figure 3.20: Path-loss measurements under different indoor and outdoor environments.

tation for communication energy when channel is noisy (high path-loss). Also, with lower BER requirement as is shown in Fig. 3.19.(b), the system performs less computation (no PD$=3$ mode is observed) and operates at smaller PA gains. Energy per frame under different environment are also shown. Finally, the end-to-end system is deployed on a mobile IoT platform and various indoor and outdoor conditions are used to evaluate the potential of the

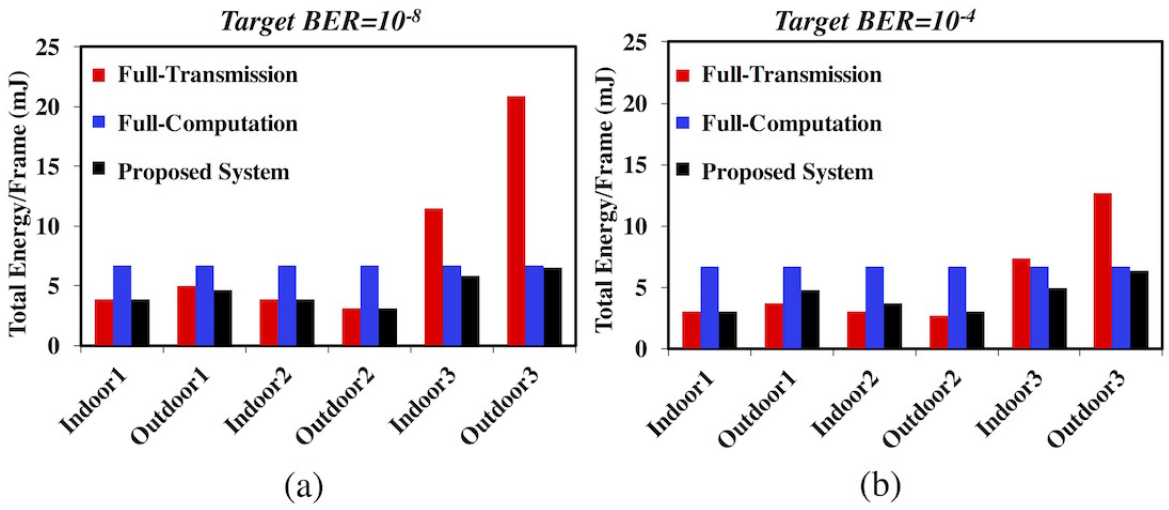


Figure 3.21: Measured total energy (average) per frame in different environments vis-a-vis static designs under BER targets of (a) 10^{-8} and (b) 10^{-4}.

design. Path-loss as a function of distance between the IoT node and the base-station for various wireless conditions are shown in Fig. 3.20. For these operating conditons, we compare the total energy/frame dissipated in the proposed system vis-a-vis "Full-Transmission" and "Full-Computation" designs. The comparative results for two BER targets are shown in Fig. 3.21. We note that the proposed system saves significant energy during run time and the optimal balance between computational energy and communication energy is obtained.

Fig. 4.29 shows the comparison with state-of-art designs on low-power wireless video applications. Previous research efforts have been focused on either (1) embedded low-power video processing [97, 98, 99], such as SRAM-FPGA based on-board object detection, or (2) adaptive wireless communication which adjusts the PA power and transmitter linearity with the dynamic wireless channel conditions [100, 101], To the best, of our knowledge this is the first reported work where the computational and communication energies are being co-optimized for achieve the highest energy efficiency. To compare the proposed system with published results, the power numbers reported are normalized to the image size (320×240), maximum TX output power (20dBm) to estimate the final metric of energy per frame. The comparison shows that the proposed system outperforms state-of-art

design by more than $2\times$.

	System/ design	*Application/algorithm*	*Embedded Processing Unit Parameter*	*Image size*	*Performance*	*Max. output TX power*	*Energy consumption*	*Estimated energy per frame*
Embedded Processing	[36]	Object Detection & tracking	V_{core}=1.2V f_{clk}= 60-160Mhz tech. node = 45nm	320x240	fps=2	N.A.	319.9 mJ/frame	>319.9 mJ/frame
	[37]	Particle detection/meter/reading/ people counting	V_{core}=1.2V f_{clk}= 520Mhz tech. node = 45nm	640x400	fps=16-48	5 dBm	17.2-22.8 mJ/frame	17.5-29.2 mJ/frame
	[38]	Particle detection	V_{core}=1.2V f_{clk}= 60-160Mhz tech. node = 45nm	640x400	fps=48	N.A.	14.12 mJ/frame	>14.12 mJ/frame
Adaptive Wireless	[39]	Video offloading	N.A.	N.A.	data rate=1.5 Mb/s	N.A.	1223 mW	162.8 mJ/frame
	[40]	Sensor node	N.A.	N.A.	BW=2.2-2.6 Ghz	14.5-23.5 dBm	165-912 mW	6.1-34 mJ/frame
Embedded Processing + Adaptive Wireless	Proposed Work	Human detection	V_{core}=1.1V f_{clk}= 100-400Mhz tech. node = 65nm	320x240	fps=10	20 dBm	4-7 mJ/frame	4.8-8.4 mJ/frame

Figure 3.22: Comparison table: The proposed system has been compared with state-of-the-art video based sensor nodes which either (1) perform "in-sensor" video processing, or (2) improve energy-efficiency of the wireless transmitter through real-time adaptation. The proposed system performs self-optimization between the computation and communication to enable the lowest power consumption in a dynamic environment.

3.7 Conclusion

This chapter presents a video IoT sensor node which performs self-optimization between the amount of computation (for human detection) and the total data volume to be transmitted. As the information content and the channel conditions change, the system tracks the minimum energy point. Hardware measurements show $4.3\times$ reduction of the total energy/frame compared to a baseline design. Comparisons with state-of-the-art video based sensor nodes, we note more than $2\times$ reduction in energy/frame.

3.8 Discussions

3.8.1 System Inefficiency

This platform has systematically proved the significance of computation and communication trade-off with respect to energy in dynamic environment. However, it consumes hundreds of miliwatts for the video surveillance task and typical battery's stored energy will drain away in only several weeks. As an practical platform instead of prototype, it lacks

energy efficiency. One major cause of the inefficiency comes from its adopted conventional IoT design methodology as discussed in the chapter I. In conventional IoT design, the data has traversed a long path towards extracted information through stacked hierarchies. From quantization, boolean logic all the way to general purpose compiler of the ADI image processing board, the energy loss is significant. At the same time, extensive unused modules, such as processing element, on-chip SRAM, peripherals and so on, has introduced large amount of static energy consumption.

3.8.2 Control Overhead

Online computation and communication trade-off requires proper control modules. This control module will inevitably introduce overhead. In this design, control overhead has not been accounted for: 1) Controller is implemented on the PC whose OS is needed to configure SDR, but in practical platform, we need much more efficient module to account to minimize area/power overhead; 2), SDR has wide and fine-grained programmability which are preferred for the proposed application/control algorithm. However, as a discrete component, the reconfiguration latency is significant in real time application and highly dynamic environment. The TX configuration overhead need to be minimized; 3), Environment modelling is at the core of wireless IoT controller. Due to model inaccuracy or environmental change, the control system will need calibration after deployment. In this scenario, controller's self-learning capability is preferred. For the proposed system, controller utilizes fixed coefficient LUT which is incompatible with online self-learning scheme.

3.8.3 DNN Computation Architecture

For state-of-art image processing algorithm, deep-neural-network (DNN) is widely applied for its superior performance in object detection, image classification and so on. In the proposed platform, such algorithm has not been explored nor optimized. For state-of-art wireless image-based IoT device, an DNN accelerator is highly demanded to handle vari-

ous applications. Further, like a general-purpose image processing unit, the DNN accelerator needs to provide programmable computation pipeline to account for various network topology/layer types.

CHAPTER 4

A WIRELESS IMAGE PROCESSING SOC ENABLING EI

As discussed in previous chapters, *EI* design with discrete components lacks the required energy efficiency and real-time controllability. To investigate the improvements gained through fully-integrated design and validate computation-communication trade-off with state-of-art customized circuit, this chapter discusses an SoC design integrating both computation and communication units that efficiently trades-off computation-communication for wireless image processing applications. This chapter is an expanded version of "A 65nm Image Processing SoC Supporting Multiple DNN Models and Real-Time Computation-Communication Trade-off via Actor-Critical Neuro-Controller" presented to 2020 Symposia on VLSI Technology and Circuits with the dissertation author as the primary author.

4.1 Introduction

The wide spread proliferation of smart sensors has led to hardware that enable edge intelligence (EI) with extreme energy-efficiencies. This decreases the volume of data that is transmitted to the cloud, thus reducing: (1) processing latency, (2) communication energy and (3) network congestion. However, this comes with an added cost of computation at the edge node [1-3] (Fig. 4.1.(a)). The cost (energy/latency) of edge computation and the cost of communication to the cloud vary widely depending on operating conditions, that include (1) information content in the data, (2) algorithm selection, (3) channel conditions (noise, path-loss etc.), (4) network size, available bandwidth and (5) resources at the cloud, as shown in Fig. 4.1.(b). We call the number of NN layers processed at the edge, processing-depth (PD). Increasing edge-computation increases PD, but reduces the volume of data to be transmitted. This not only provides an opportunity to efficiently configure computation and communication blocks but also trade-off between computation and communication in

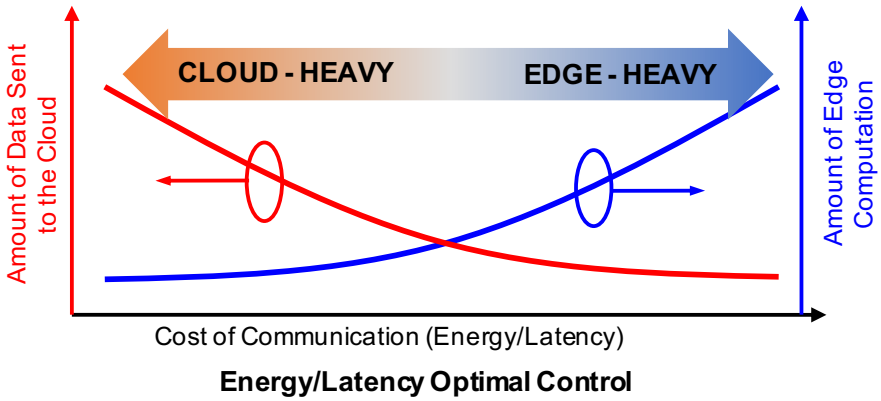
real-time to meet system targets.

Wireless image processing is among the most demanding use-cases for optimal computation-communication trade-off. On one hand, image processing is at the core of many important applications, such as surveillance, authentication, recognition, behavior analysis and so on. On the other hand, the high-dimensional data volume together with extensive computation (deep neural-networks and etc.) have brought about significant challenges to resource-constrained wireless IoT platforms. Both facts have motivated us to investigate chip-level solutions to addresses various wireless image processing challenges with systematic optimization and state-of-art circuit techniques.

This chapter presents a 65nm wireless image processing SoC for real-time computation-communication trade-off on resource-constrained edge devices. The test-chip includes (1) an all-digital, near-memory, reconfigurable and programmable neural-network (NN) based systolic image processor at 1.05TOPS/W (peak), (2) a digitally-adaptive RF-DAC based transceiver with Tx energy-efficiency of 768pJ/b and (3) a mixed-signal, time-based, actor-critic neuro-controller with compute-in-memory (CIM) and in-place weight updates that provides online learning and adaptation at 0.59pJ/MAC for efficiently controlling the computation, communication blocks separately as well as jointly.

4.2 System Analysis

Conventionally IoT image processing schemes either directly transmit captured image to the back-end server or process end-to-end algorithms locally without data exchange. As mentioned in previous section, both schemes lacks environmental awareness and systematic optimization. The smart wireless image processing scheme proposed is shown in Fig. 4.2.(a). There are three major building modules: pipelined computation, adaptive communication and optimal policy control. Such a system optimizes programmable system targets (y_T) according to dynamic sensed variables (u_D) through various control knobs (CTRL). The detailed variables are denoted in Fig. 4.2.(b). A systematic overview and



(a)

Sources of Variations		
Local Environment	***Wireless Channel***	***Sensor Network***
• PT Variation • Algorithm Selection • Information Content	• Channel Noise • Path-Loss • Interference	• Network Size • Available Bandwidth • Back-End Resources

(b)

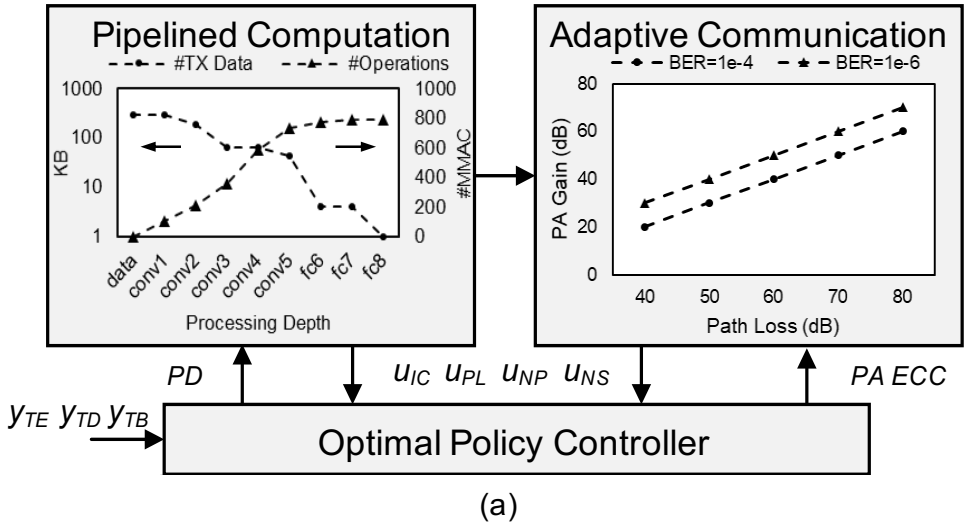
Figure 4.1: Edge computation and cloud communication trade-off.

analysis of the three modules are discussed in this section.

4.2.1 DNN Image Processing Pipeline

Deep neural-network (DNN) is the state-of-art image processing framework and has even achieved performance superior than human in certain applications, such as image recognition, object detection and so on. Compared with shallow multi-layer perception, DNN usually has extensive cascaded/parallel convolution layers to extract features and several fully-connected layers at the end to separate feature space. Further, people have looked into pruning techniques to sparsify neural-network to maximally reduce computation/storage bottleneck for embedded system.

To understand DNN processing, 4 widely applied network topologies (AlexNet, GoogleNet, SqueezeNet and VGG16) are analyzed. Fig. 4.3 shows output data volume and accumulated number of operations at certain layers in these DNNs. We have observed a monotonically



(a)

Design Targets (y_T)	Sensed Variables (u_D)	Control Knobs (CTRL)
y_{TE}: Target Energy y_{TD}: Target Latency y_{TB}: Target BER	u_{PL}[1:0]: Path-Loss u_{NP}[1:0]: Noise Power u_{NS}[1:0]: Network Size u_{IC}[1:0]: Information Content	PD[1:0]: Processing Depth PA[1:0]: PA Gain ECC: Error Correction Code

(b)

Figure 4.2: Self-optimizing platform.

decreased output data volume and monotonically increased computation workload with respect to deeper DNN processing depths (PD). It means that DNN framework is inherently compatible to act as an computation-communication trade-off scheme: shallow PD for edge computation savings and deep PD for data communication savings depending on dynamic communication cost.

At the same time, each DNN topology has its own computation characteristics with respect to computation workload, data transfer patterns, layer specifications and so on. The proposed DNN computation pipeline as a processor should feature not only any particular DNN, but DNNs in general to adapt to wide future use-cases. The optimization is implemented on both macro and micro levels: (1), pipeline is reconfigurable to account for workload distribution between PDs and maximize local intermediate data utilization across

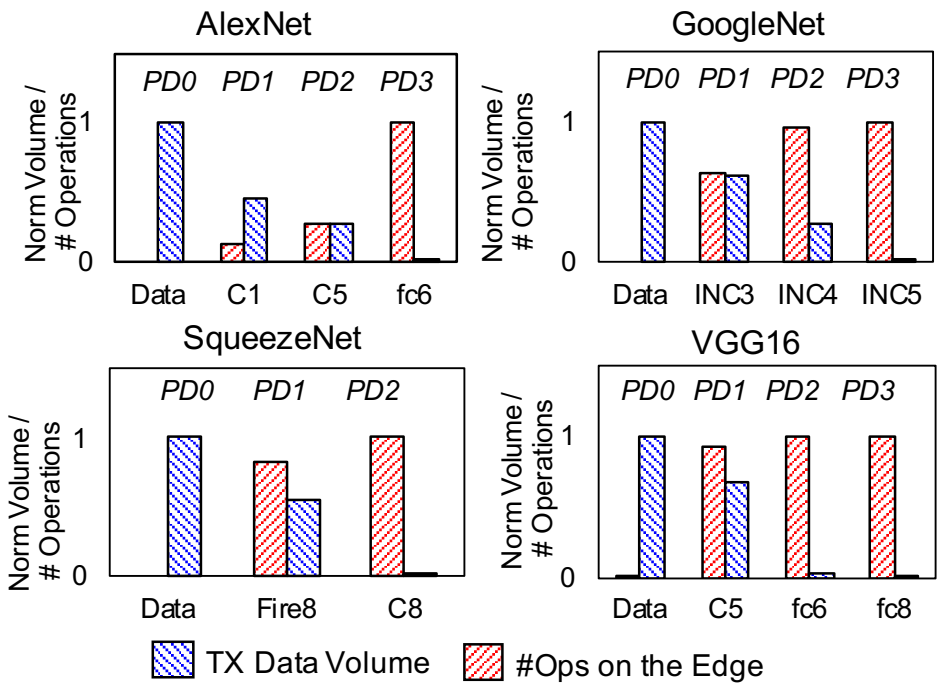


Figure 4.3: Output data volume and accumulative number of computations across layers for various DNN architectures.

DNNs; (2), the processing element in the pipeline is able to reconfigure for layer-wise optimizations, such as convolution layers, fully-connected layers, sparsely-connected layers and so on. The detailed implementation will be discussed in following sections.

4.2.2 Adaptive Communication

Wireless environment is highly dynamic. To guarantee data transmission (Tx) accuracy, transceivers are conventionally designed for the worst case, which becomes a major power consumer for the edge system. To mitigate communication energy bottleneck, adaptive transmission has been extensively explored. By monitoring dynamic wireless channel conditions, the Tx control knobs are tuned accordingly to provide marginal performance thus preserve energy consumption.

An example of adaptive communication is illustrated in Fig. 4.4. In the first case, channel suffers from severe path-loss (70dB) and data accuracy is critical (BER10^{-8}), thus output PA power will be tuned to high gain resulting in significant Tx power. On the

contrary, when channel loss is moderate (30dB) and transmission data error tolerance is high (BER10^{-4}) as in the second case, the transceiver can save to up to 100× Tx power by properly lowering PA gain in this example.

In hardware adaptive transceiver design, we would like to incorporate more programmable knobs to provide high degree of freedom. Meanwhile, efficient on-chip transceiver (TRx) implementation is highly desired for responsive Tx control. The adaptive transceiver details are discussed in following sections.

4.2.3 Optimal Control

Besides computation pipeline and adaptive communication, it is crucial to optimally control the two modules independently as well as the integrated system. The controller will take design targets and sensed variables as input and dynamically choose control knobs as output. In a complex environment, both input/output dynamic range and variable size will be large, It will consequently lead to significant policy search space and make real-time control more challenging. Further, for complex environment, it is difficult to model the system accurately. The devices have to be able to calibrate off-line trained/modeled policy and learn in the deployed environment over time. It requires thorough investigations into the choice of control scheme.

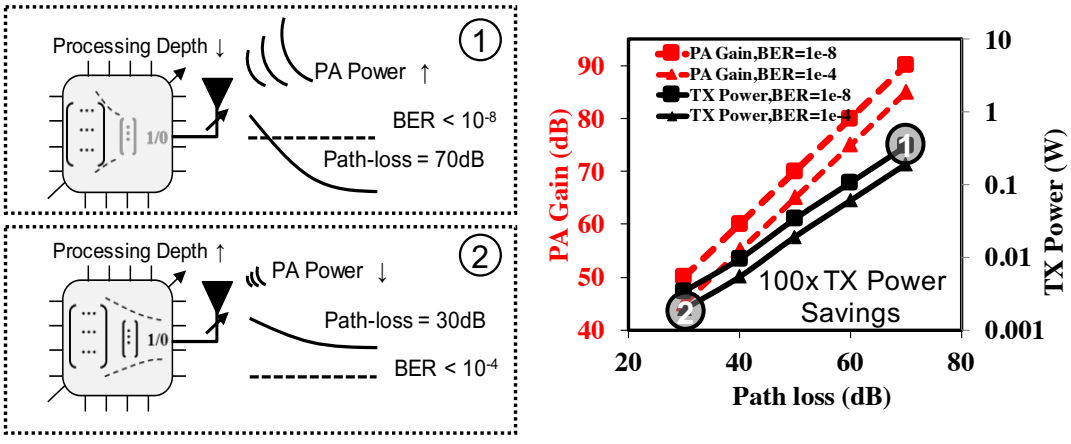
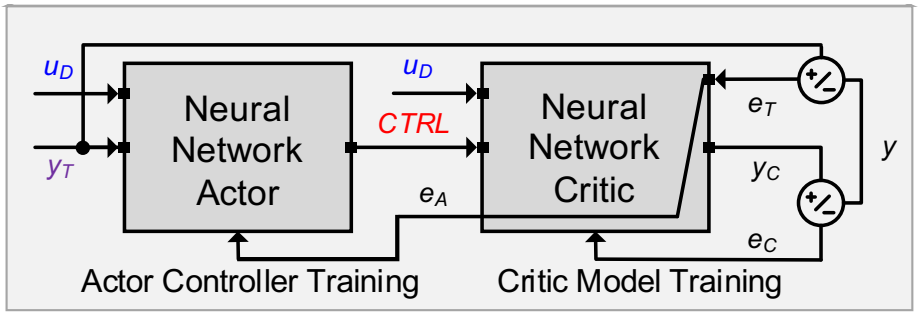


Figure 4.4: Adaptive communication example that PA gain adapt to path-loss and BER requirements to preserver energy.

One straightforward solution is to offload control to the cloud. With immense computation resources at the back-end, the control can handle highly complex environment with respect to processing and will be able to calibrate the model at the same time. However, there is a delayed control with respect to locally sensed variables. This delay may result in platform energy waste or even consecutive transmission failures in the worst-case with over-optimistic control choice. Alternatively, we may choose embedded look-up-table (LUT) as a controller implementation. We are able to use input as index to find optimal policy immediately. However, LUT lacks learnability. To improve learnability, we may choose neural-network as an emulator for the platform. By emulating control knobs together with sensed variables, we can easily locate optimal policy. The problem from a hardware perspective is that exhaustive search is required. Both energy and delay overheads make such a scheme less preferable.

CTRL Model	Real-time Control	Real-time Learning	Policy Search
Remote CTRL (cloud)	no	yes	/
Look-up table (LUT)	yes	no	index
NN Emulator	yes	yes	exhaustive
NN-based Actor-Critic	yes	yes	single inference

(a)



(b)

Figure 4.5: (a) Neural-network-based actor-critic controller; (b) optimal policy control scheme comparisons.

To address all the problems mentioned above, we have chosen neural-network-based actor-critic (AC) control scheme. It has an actor neural-network and critic neural-network, one for making decisions and one for system emulation. In run-time, the actor picks optimal control knobs in a single shot with sensed variables and design targets; and the critic emulates chip performance with sensed variables and selected controls. During training, emulation errors are collected to calibrate critic neural-network, while the target errors at the output of critic controller are back-propagated through critic controller as control errors to train actor neural-network. The AC-controller is able to provide both real-time and learnable optimal control. The control scheme comparison is shown in Fig. 4.5.(a) and data flows are shown in Fig. 4.5.(b).

In actual hardware implementation, we expect our controller to be as efficient as possible, for both inference and learning. The implementations details will be discussed in the following sections.

4.3 System Architecture

The SoC architecture is shown in Fig. 4.6. There are three major blocks designed for DNN computation pipeline, adaptive communication and optimal control respectively:

1. **PE Spatial Array**: A 3-by-3 processing element (PE) array with reconfigurable interconnections between PEs to account for various DNN architectures. Each PE has 8 threads (each thread with an ALU, a 1KB SRAM, and a shift register). PE is also reconfigurable for optimized layer operations.

2. **Adaptive Transceiver**: On-chip digitally reconfigurable channel-aware transceiver with programmable power amplifier (PA) gain, data rate and error correction code mode.

3. **Actor-Critic Controller**: A neuro-based actor-critic controller. Both controllers are 2-layer neural-network with each layer implemented with 10-by-10 compute-

in-memory (CIM) module.

Besides the major building blocks, the SoC has also included an 8KB frame buffer to store input image, a pre-processor to infer frame difference, data/instruction caches to store temporal data/instructions, a scan chain and a decoder.

The SoC interfaces with camera, power supply and management unit, optional external DRAM and programmable interface. The SoC will be remotely connected with cloud server for data exchange through on-chip transceiver.

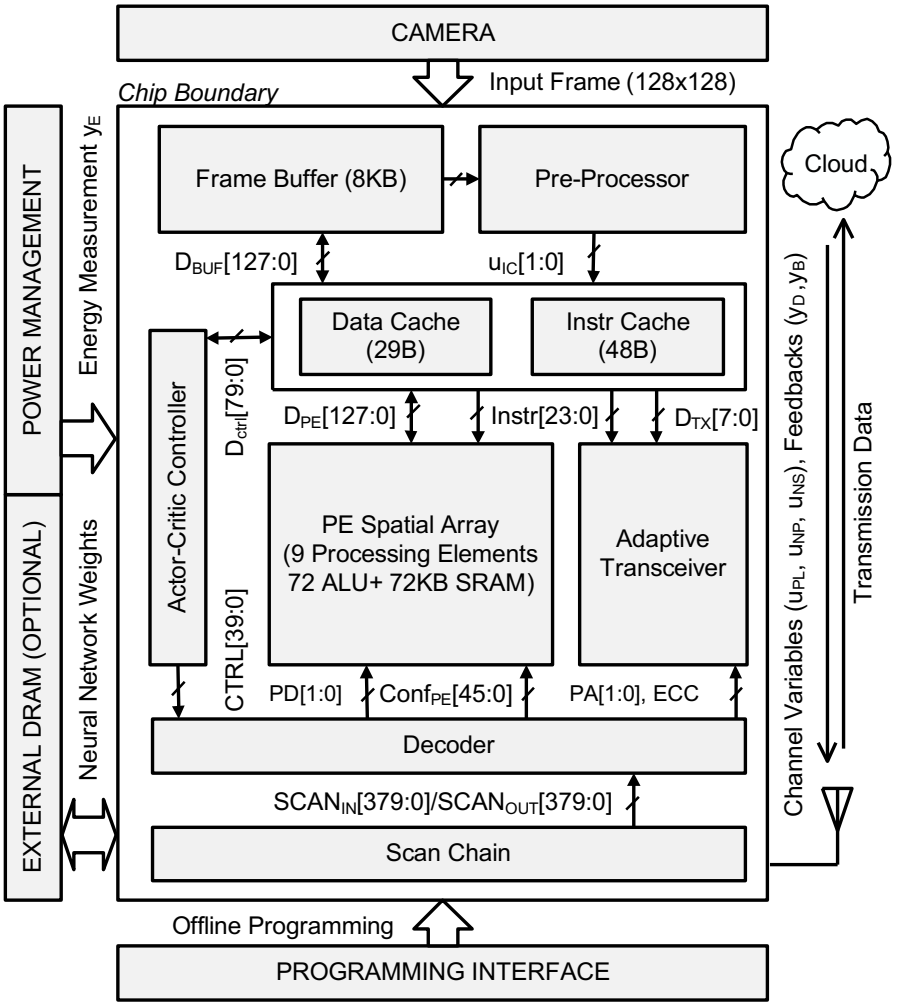


Figure 4.6: System architecture

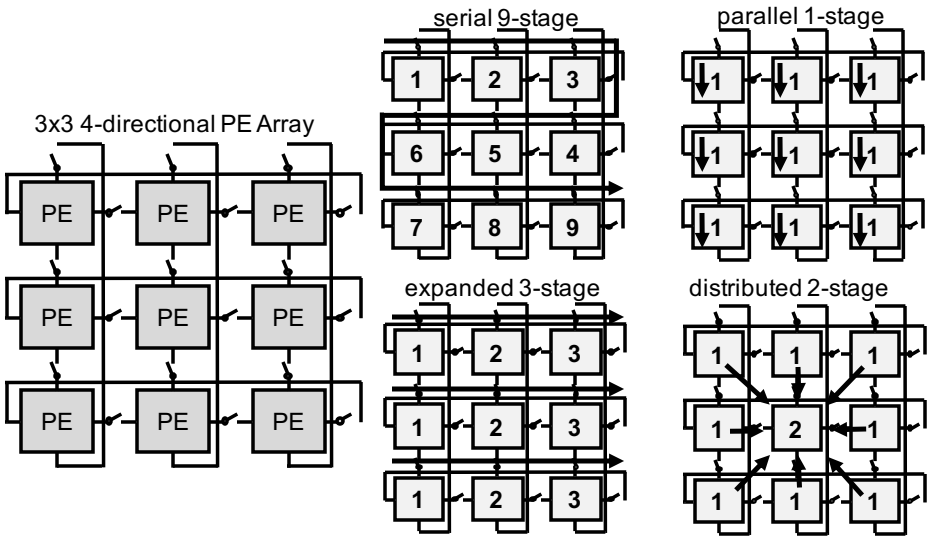


Figure 4.7: 3x3 PE spatial array for reconfigurable DNN pipelines.

4.4 Circuit Design

4.4.1 Reconfigurable PE Spatial Array

The PE spatial array has 9 PEs and the PEs are placed in a 3-by-3 configurations as is shown in Fig. 4.7. Each PE is able to reconfigure its input to any of the outputs of 4 adjacent PEs. At the same time, each PE can bypass the data so that one PE's output data can directly reach any other PE. By controlling each PE's interconnection and bypass status, the PE array can be easily reconfigured for various pipeline topologies depending on workload distribution and data-flow pattern. For example, in a deep pipeline where workload is evenly distributed and data is sequentially passed on to next stage, the PE array can be reconfigured to support up to 9 stage serial pipeline. On the contrary, if a workload is highly parallel and there are minimal data exchange between computations, the array can also be reconfigured as parallel 1-stage pipeline. And it can also form any pipeline between 1 stage and 9 stages as is shown in Fig. 4.7.

PE in the array (shown in Fig. 4.8) includes 8 threads, and each thread consists of the following sub-blocks:

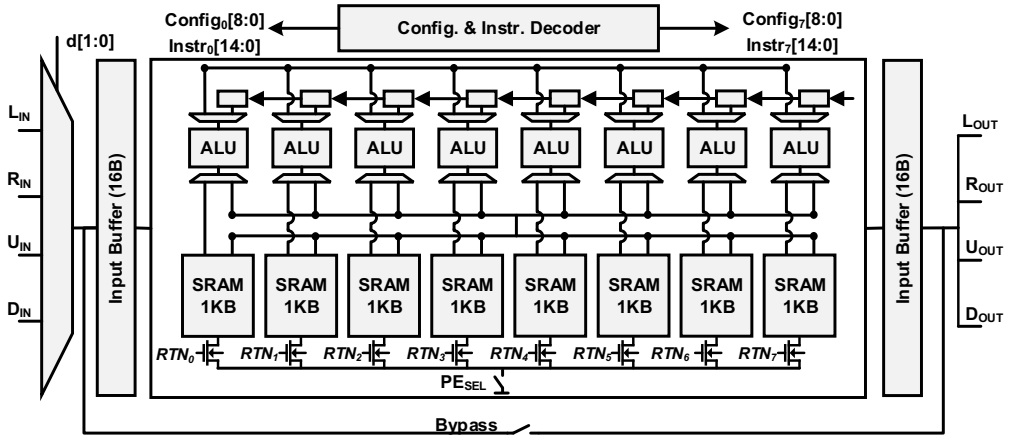


Figure 4.8: Reconfigurable PE for various DNN layers.

1. **Arithmetic Logic Unit**: ALU's inputs are connected with two 2-to-1 multiplexers. Input A is able to select between (1), the data on the global bus at the output of input buffer shared by all threads and (2), the data stored in local shift-register. Input B is able to select data between (1), data on global bus at the output of any particular SRAM in the memory bank and (2), data read from local SRAM.

2. **Retention-enabled SRAM**: The SRAM output is both connected to the ALU within the same thread and a global bus shared with all SRAM blocks in the PE. Further, to reduce static power consumption of un-accessed SRAM, the PE have full control to put any SRAM blocks into retention mode.

3. **Shift-register**: The shift registers are connected to other shift-registers in its neighbouring threads. The first shift-register is connected to input buffer. The shift buffer chain will work as a FIFO register array when needed and push input one data in at each clock cycle.

With proper configurations, the PE is able to optimize for various DNN layer types who differ in computation pattern and memory usage. In particular, the energy-efficiency and throughput of fully-connected layers, convolution layers and sparsely-connected layers are most important features to optimize in DNN acceleration.

Fully-connected configuration is depicted in Fig. 4.9.(a). All threads work in parallel. The ALU selects one input from global input bus and another input from local memory which stores weights. Each thread acts as an output neuron as shown in Fig. 4.9.(a). By feeding sequential input data to PE, a maximum of 8 output neurons will be computed at the same time through multiplication and accumulation (MAC) operations on parallel ALUs. The input sharing minimized input data access and computation parallelism improved throughput. It should be noted that fully-connected layer's computation is essentially parallel vector product between input data vectors and weight vectors. As a result, such configurable also applies to 1-by-1 filter kernel in GoogleNet, SqueezeNet and MobileNet.

Convolution configuration is described in Fig. 4.10.(a). All ALUs compute MAC in parallel where one input from local shift-register (input data) and the other from global memory bus (weight). During the computation, only the SRAM stores the weight will be active, while all others in retention. All threads compute convolutions with the same kernel of diverse portions in the input data array (shown in Fig. 4.10.(b)). By feeding sequential input data to PE and shift the data, a maximum of 8 convolutions can be processed in each clock cycle. Further, as filter weights are shared with all threads, un-accessed memory sub-banks can be put into retention to save static memory energy expenditure.

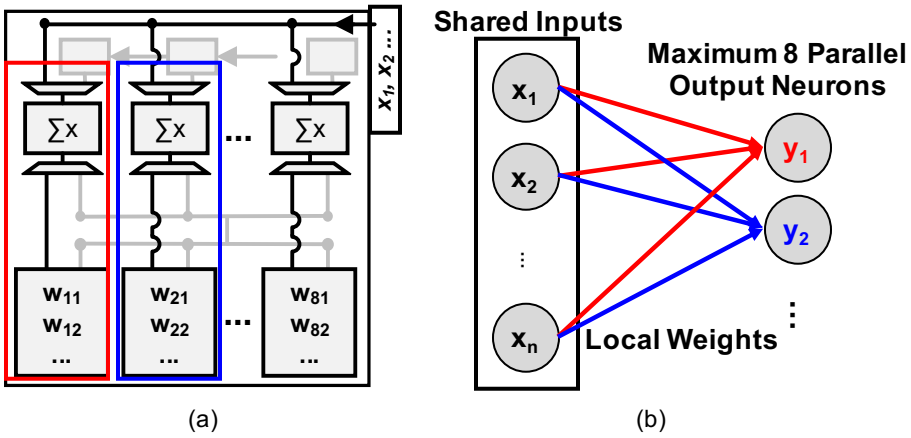


Figure 4.9: Fully-connected layers (a),configurations and (b),computation.

Sparsely-connected layer's PE configuration is shown in Fig. 4.11.(a). The threads will be assigned to either MAC or accumulation tasks respectively. The ones assigned to MAC tasks will collect input data from global data bus in a pipelined manner. The thread to perform accumuation will be acting as an index accumulator to compute which input should be fetched for computation. The accumulator will read from its local memory of index difference and accumulate them for actual index. Un-used SRAMs are put into retention mode.

4.4.2 Reconfigurable RF-DAC Tx and ULP OOK Rx

To have energy-efficient communication with an external Hub, a digitally reconfigurable, data-rate and channel-aware transceiver (Fig. 4.12) is designed on the same SoC that demonstrates the effectiveness of computation-communication trade-offs through adapting to various data rates and channel conditions. The input data for test purpose comes from the PRBS generator. Alternatively, real-data from the on-chip compute units are utilized as input, which can be selected by the baseband select mux. The data rate control is achieved by changing the clock rate from 40kHz to 10MHz in 256 steps using an 8-bit control. The ECC can be enabled by one control bit that turns on [8,4] Hamming codes. From the digital baseband, 3 bits of I amplitude control and 3 bits of Q amplitude control controls how many

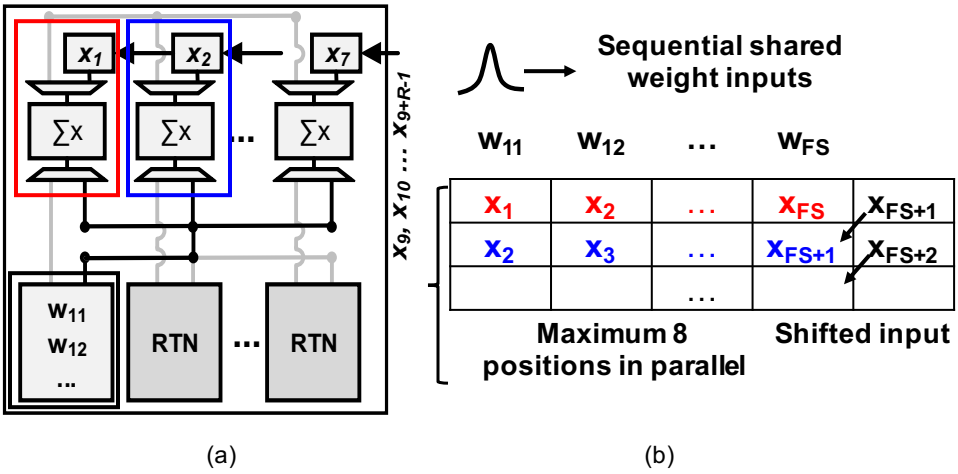


Figure 4.10: Convolution layers (a), configuration and (b), computation.

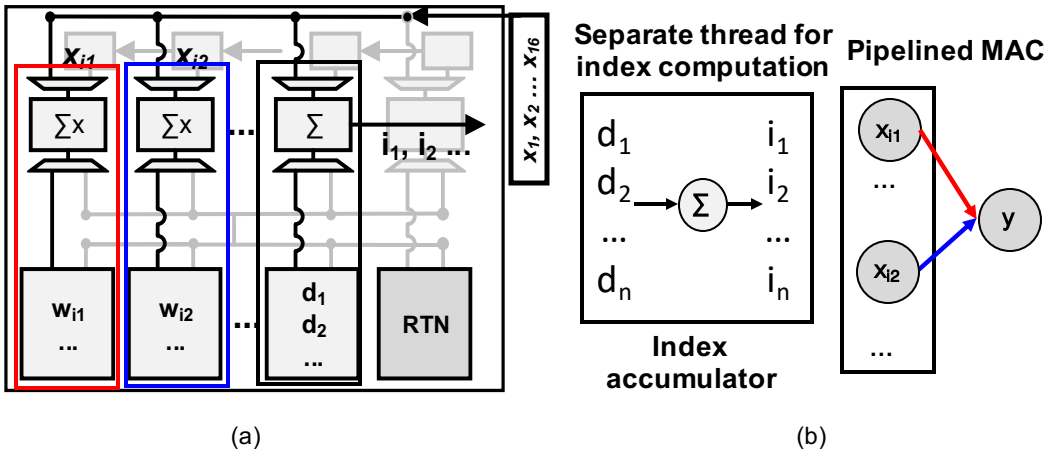


Figure 4.11: Sparsely-connected layers (a), configuration and (b), computation.

legs of the RF-DAC would be turned on, while 2 bits of LO control generates appropriate local oscillator phases.

The power delivery subsystem consists a reconfigurable RF-DAC based PA (Fig. 4.13.(a)) and a tapped capacitor matching network with reconfigurable capacitor banks. The RF-DAC based PA combines the DAC, mixer and PA operations in a single module through a digital-friendly architecture that switches on or off different legs of the module, and can support higher order modulation schemes such as 16-QAM or 64-QAM. The output power control is achieved using 3 bits that alters the capacitor banks present in the matching network. As mentioned earlier, 3 bits of I-path amplitude control and Q-path amplitude control determines how many legs in the I-path and Q-path would be turned on.

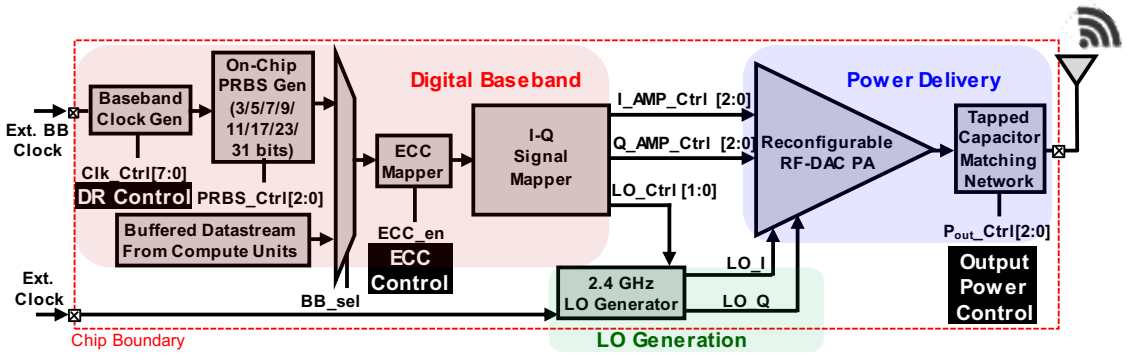


Figure 4.12: Adaptive transmitter circuit.

The 2.4GHz LO generation (Fig. 4.13.(b)) for both I and Q paths is performed by an on-chip LO generator, as shown in the right-hand side of this slide. 4 LO phases are selected based on the 2bit LO control as obtained from the I-Q signal mapper.

For the design of the matching network, the effect of finite Q of the on-chip inductor is considered, which modifies the well-known formula obtained in the ideal scenario that considers an infinite Q. 5 different values of C1 and C2 are considered and are put on chip as a part of two different capacitor banks that cover a matching network impedance from about 100 ohms to about 1600 ohms. While designing the matching network, the effects of pad capacitance, bond-wire inductance and pcb capacitance are also considered, and the effect of the additional capacitances are included in the value of the on-chip capacitor, C1. The capacitor matching design is described in Fig. 4.14.(a-b).

Along with the transmitter, we also have an ultra low power OOK receiver (Fig. 4.15.(a)) on the same chip that captures control signals from a nearby base station to achieve closed-loop control on the 8 clock control bits, 1 ECC control bit and 3 Pout control bits. The receiver consists of 2 stages of RF LNA, a differential to single ended converter, or D2S, an envelope detector, or ED, 2stages of baseband VGA and a baseband comparator.

For the envelope detector (Fig. 4.15.(b)), a 4-stage gate biased structure is used which increases the output voltage by 4X as compared to the 1stage envelope detector, thereby compensating for the loss incurred during envelope detection. The SNR, however, remains

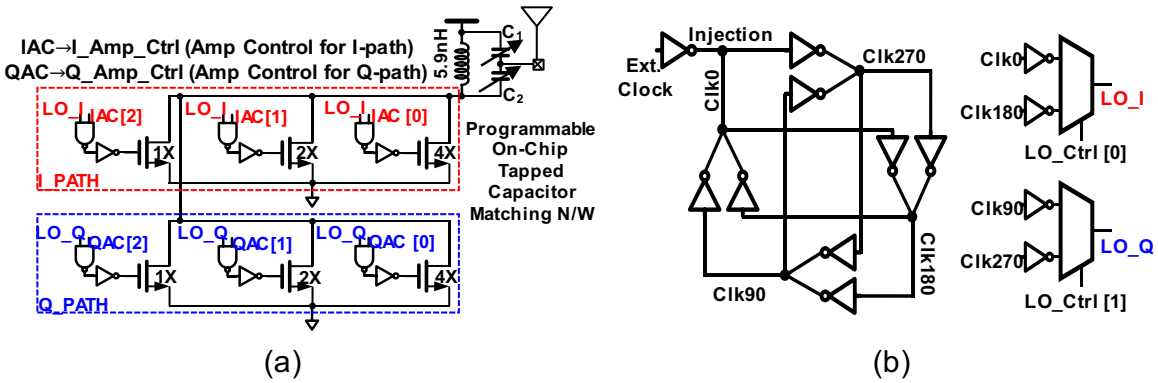
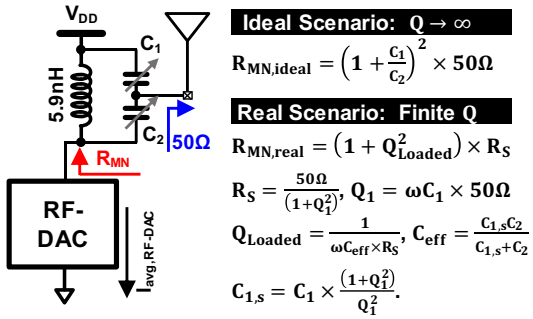


Figure 4.13: TX programmable (a), modulation circuit, (b), clock synthesis circuit.



(a)

C_1 (pF)	C_2 (pF)	$R_{MN,ideal}$	$R_{MN,real}$
0.75	2	95Ω	117Ω
0.75	1.5	112.5Ω	152Ω
1	1	200Ω	288Ω
1.5	0.75	450Ω	607Ω
2	0.5	1250Ω	1600Ω

(b)

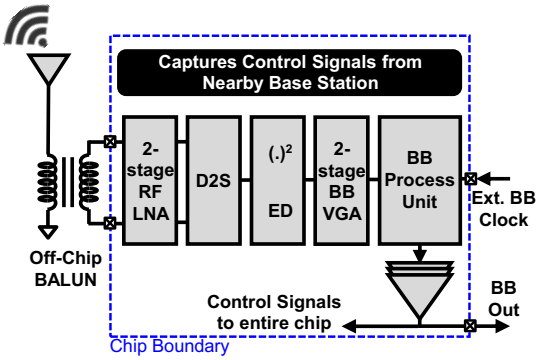
Figure 4.14: Capacitor matching.

constant as we increase the number of stages.

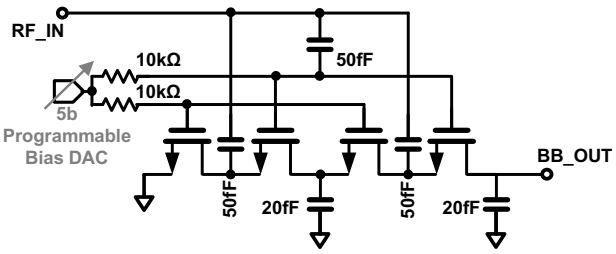
4.4.3 NN-based Actor-Critic Controller

The large control space across computation and communication is learnt using a low overhead (5% power, 2.5% area) actor-critic NN (AC-NN) controller (Fig. 4.16). The AC-NN takes both design targets and sensed variables as inputs and learns to optimally control the control knobs. These are listed in Fig. 4.2.

The controller features 4 10 x 10 memory sub-banks with time-based compute-in-memory modules. During inference, digital to time converters (DTCs) allow pulse width modulated word-lines (WLs) (input signals) to be turned on sequentially such that the



(a)



(b)

Figure 4.15: Receiver circuit design.

falling edge of one row triggers the rising edge of the next. The partial products are accumulated on the BL as long as VBL is greater than a threshold (VL) to avoid read disturb. However, if the operands are large and VBL reaches VL then the process is stopped, the ADC converts to a 6b word, the BL pre-charged and the sequence restarts. The differential bitcell and ADC allows both positive and negative weights by discharging either BL (positive) or BL_{bar} (negative). The thermometer encoding of data enables a weight update to be a left or a right shift (sign of the update), and that the duration of shift process (magnitude of update) is controlled by the DTC. The array can be read both row as well as column-wise providing a seamless design for transposing the weight matrix during back-propagation. This also enables in-place online learning without requiring reads and write-backs (baseline designs).

The thermometer encoded weight storage unit consists of 8 sequentially connected 1-b storage cells, control logic and pull up/pull down transistor at the edges as is shown in Fig. 4.18. The update is fulfilled by one single time pulse generated by DTCs and the magnitude of update is controlled by duration of the time pulse. For example, to increase magnitude, the pull-up transistor is enabled and propagate '1' to the right for certain amount of time. Bit cells will flip one by one. After the propagation, the storage element enters retention mode to store the new weight. By controlling time pulse duration and bit shift direction, stored weight can be updated without leaving the storage unit. To enable data

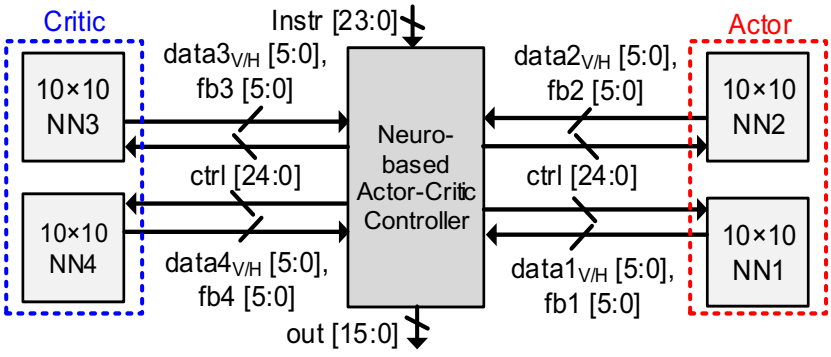


Figure 4.16: NN-based actor-critic controller circuit.

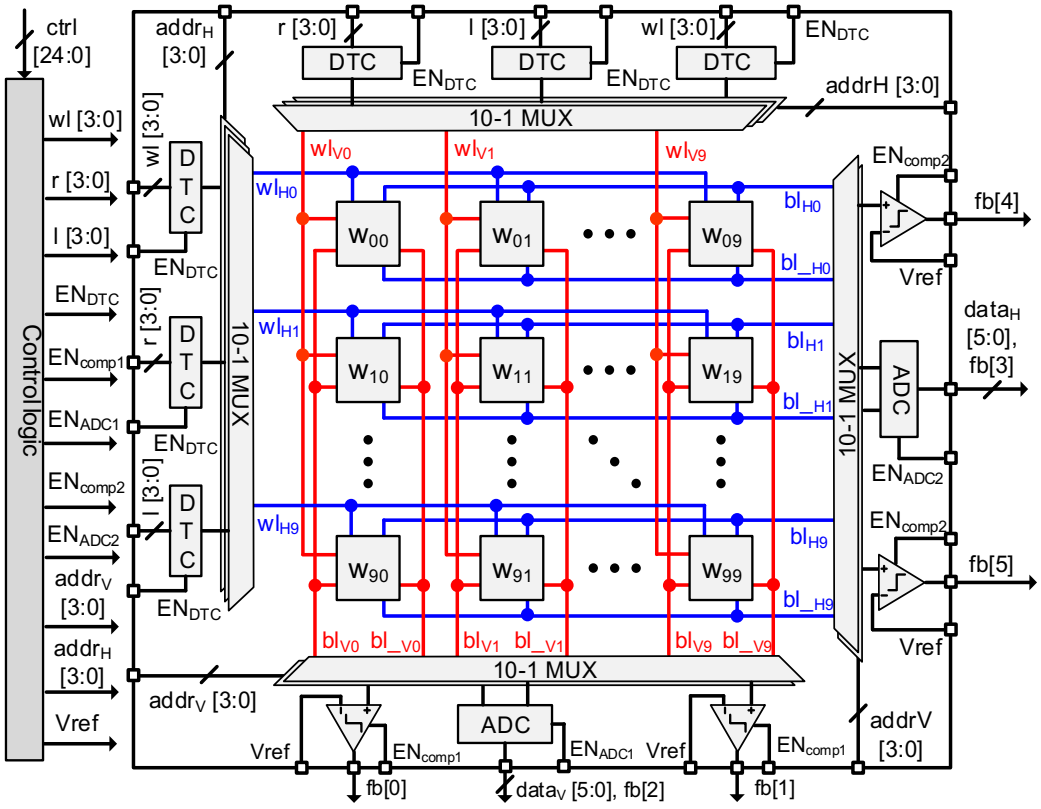


Figure 4.17: 10-by-10 compute-update-in-memory block circuit.

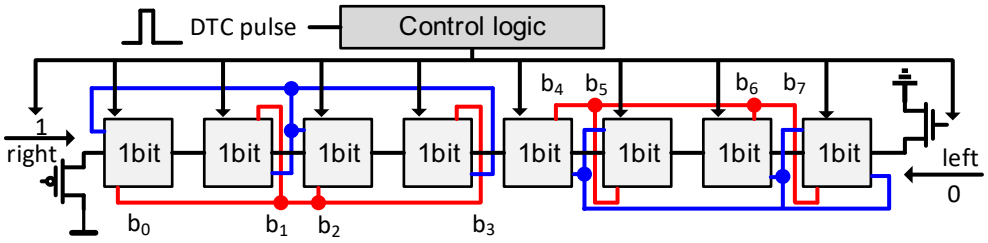


Figure 4.18: 8b thermometer-encoded memory cell.

transfer and retention, the bit cell design is described in Fig. 4.19.(a). Bit cell configurations for moving left, right and retention mode is shown in Fig. 4.19.(b). Compared conventional latch, it also includes transposed bit-line accesses and an additional transmission gate to provide bi-directional propagation.

According to the data distribution simulation, more than 90% results fall in 6-bit range while the worst case requires 8b resolution ADC. As we expect data conversion to be a major energy consumer, we decided to implement an adaptive A/D conversion scheme that

uses 6b resolution ADC for optimized energy efficiency but still supports 8b output. The circuit diagram and data simulation distribution are shown in the Fig. 4.20. We choose 6b capacitor-based SAR ADC and share the capacitors with bit-lines. This improves the dynamic range and embeds the sampling process of ADC into the compute cycle. The ADC connect with weight storage elements via a 10-1 multiplexer, and two additional comparators detect potential read disturbance. In addition, the monolithic switching procedure of ADC further reduce the energy.

The timing diagram in Fig. 4.21 illustrates the adaptive A/D conversion scheme.The bit-lines are pre-charged first. In most cases, the 10-by-10 vector multiplication is completed before conversion. However, when the intermediate sum of product gets close to the ADC's dynamic range which may cause a read disturbance on the storage cell, the computing cycle is stopped, and ADC starts to convert the bit-line voltage to digital output. After conversion, bit-lines are pre-charged again and continue computing the remaining cells. When all cells are computed, the outputs are accumulated to get the result.

4.5 Measurements

The test chip is fabraicated in 65nm technology with a total area of 5mm^2. The chip die photo and characteristics is shown in Fig. 4.22.

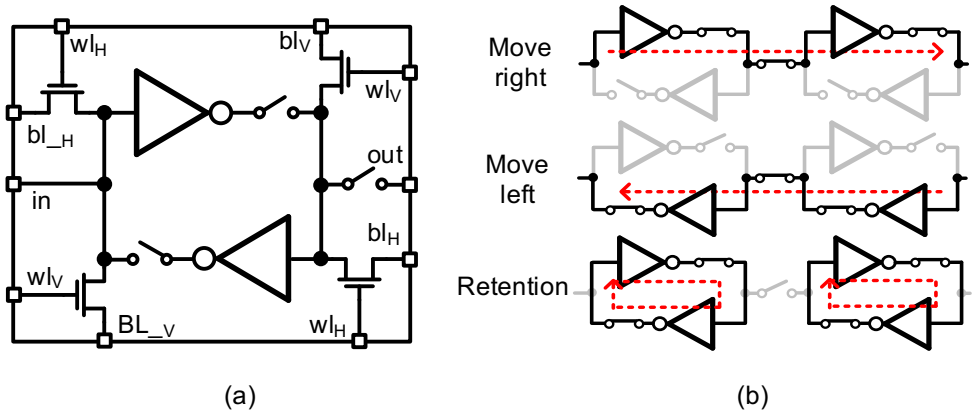


Figure 4.19: Bit cell circuit.

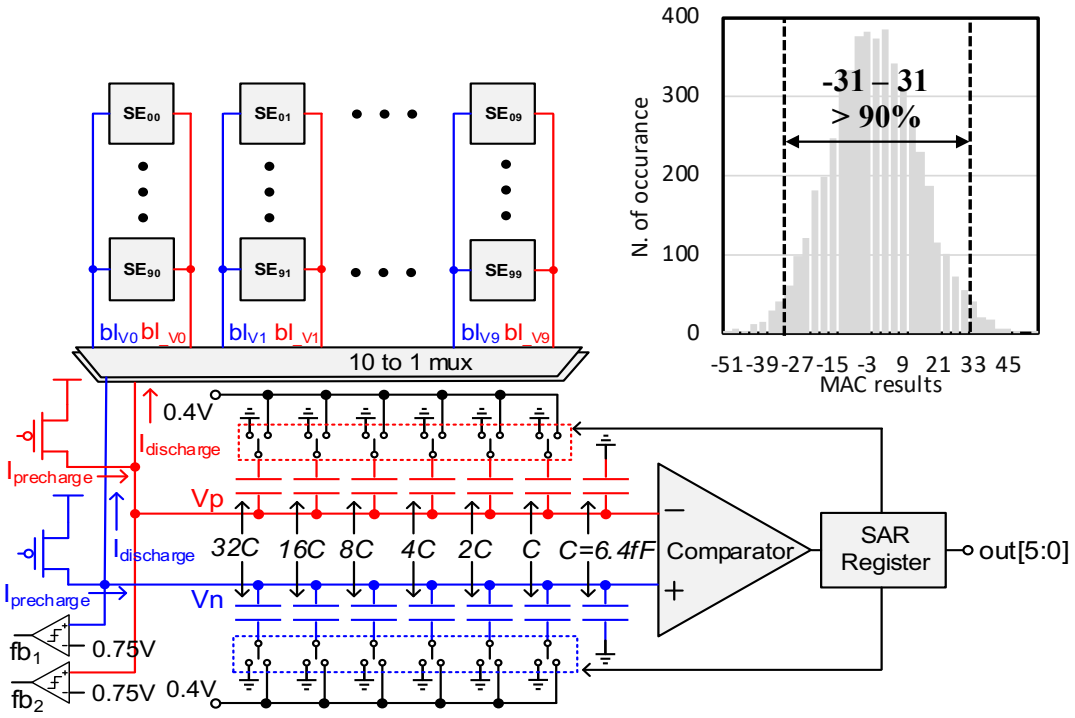


Figure 4.20: 6b ADC design and data distribution.

The measured power-performance of the processing engine (Fig. 4.23) shows VMIN of 0.5 V and FMAX of 760 MHz. Peak arithmetic energy-efficiency of 1.05 TOPS/W (0.43 TOPS/W, 0.18 TOPS/W) is measured for CONV (FC, sparse) networks at 210 MHz (0.575 V). With proposed weight sharing scheme in PE's convolution configuration and fine control of un-accessed SRAM retention mode, computation-centric convolution operation has achieved sub-pJ efficiency per operation by minimizing unnecessary memory usage.

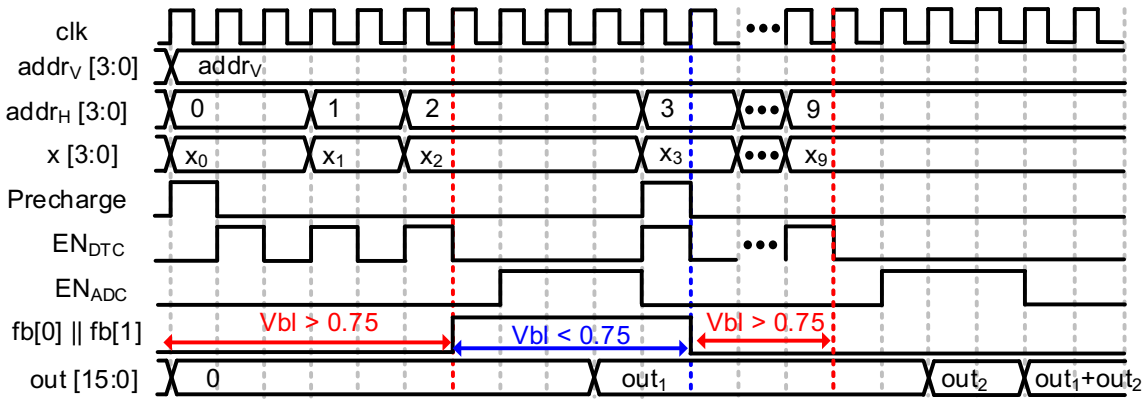
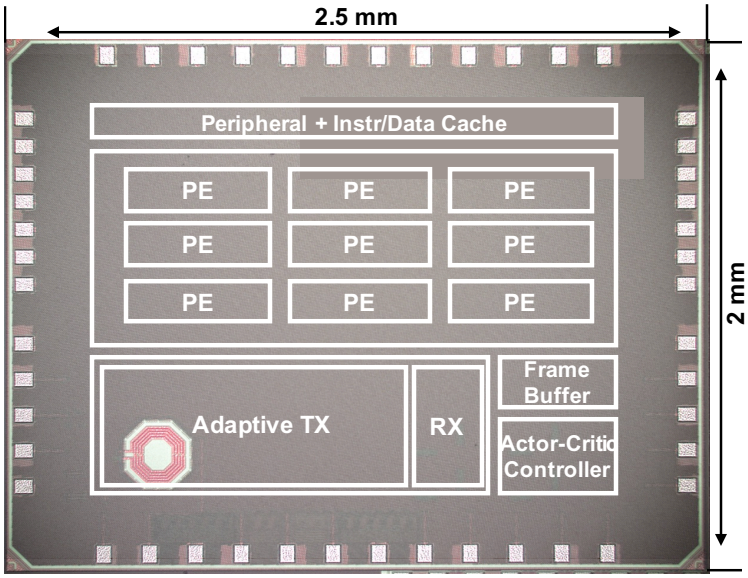


Figure 4.21: Bitline MAC timing diagram.

The RF subsystem, shows a maximum Tx efficiency of 30.3% at -0.3 dBm, with back-off efficiencies of 19.2% (7.8%) at -6.5 (-13.7dBm) with QPSK. At 1 Mbps, the Tx energy efficiency is 768 pJ/bit with 1 V supply (-0.3 dBm output power. The measured energy-efficiency for the OOK Rx is 207 (124) pJ/bit at 1 (0.8) V supply, with a sensitivity of -72 dBm for a BER of 10-3 at 1 Mbps. An [8,4] Hamming Code on the Tx improves the sensitivity to -78 dBm but halves the number of information bits.

The oscilloscope capture of neural network 10-by-10 CIM block bitline discharge is shown in Fig. 4.25. By providing 1-3 unit time worldline voltage pulse, bitline discharges proportionally with constant weights.

To investigate the computation accuracy of CIM block, we have applied random inputs to the controller at measured output result for each bitline (Fig. 4.26). First, we can observe that more than 95% of final results are within -40 to 40 range. Further, before digital



Chip Characteristics	
Technology	65nm 1P9M CMOS
Die Area	2mm*2.5mm
Testing Interface	Chip-on-Board
Pin Count	48

Figure 4.22: Chip die phot and characteristics.

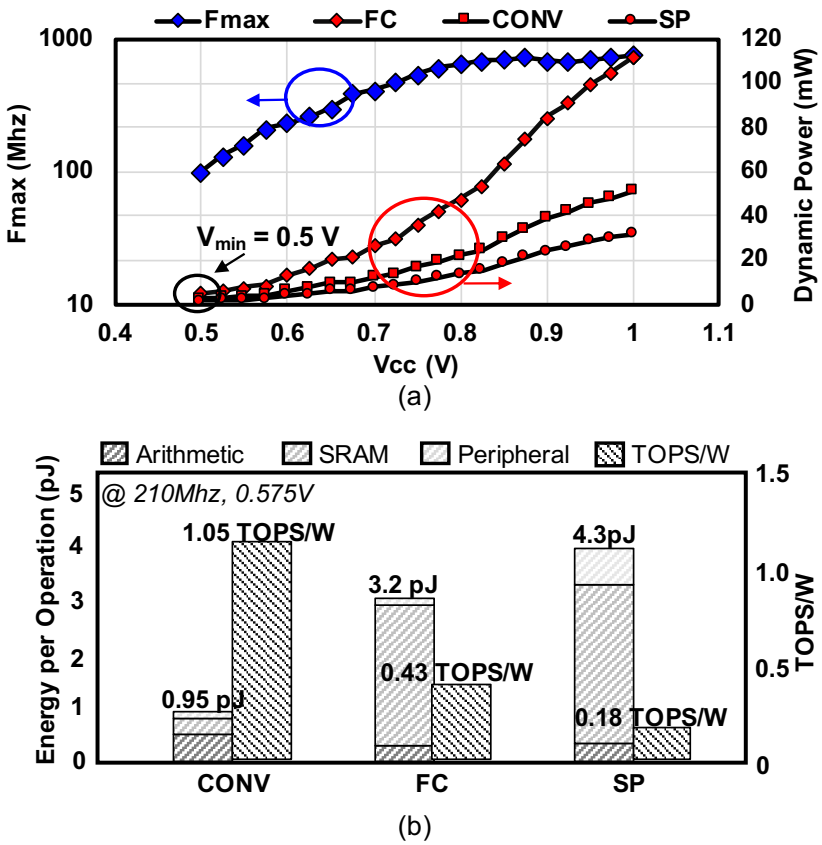


Figure 4.23: Measured (a),computation pipeline frequency/power characteristics and (b), energy consumption per operations for various layers.

compensation, we find average error increase with final computation result. That means non-linearity errors accumulates on bit-line. An average of 1.6 error is measured. After compensation, the error is largely reduced, especially in computations where final results are significant. Average error after compensation is around 0.6.

The measured performance of the neuro-controller is shown in Fig. 4.27. The CIM consumes a measured 305.2 pJ (training) and 156.8 pJ (inference) at 0.7V with less than 0.6lsb of non-linearity error. The peak measured energy efficiency is 0.59 pJ/MAC and 0.4 pJ for each weight update which are 2.2x and 4.75x lower than a digital counterparts (simulated).

The full system is deployed and neuro-controller is allowed to learn online from emulated signals from the cloud and energy meters. Then it is tested for varying noise power

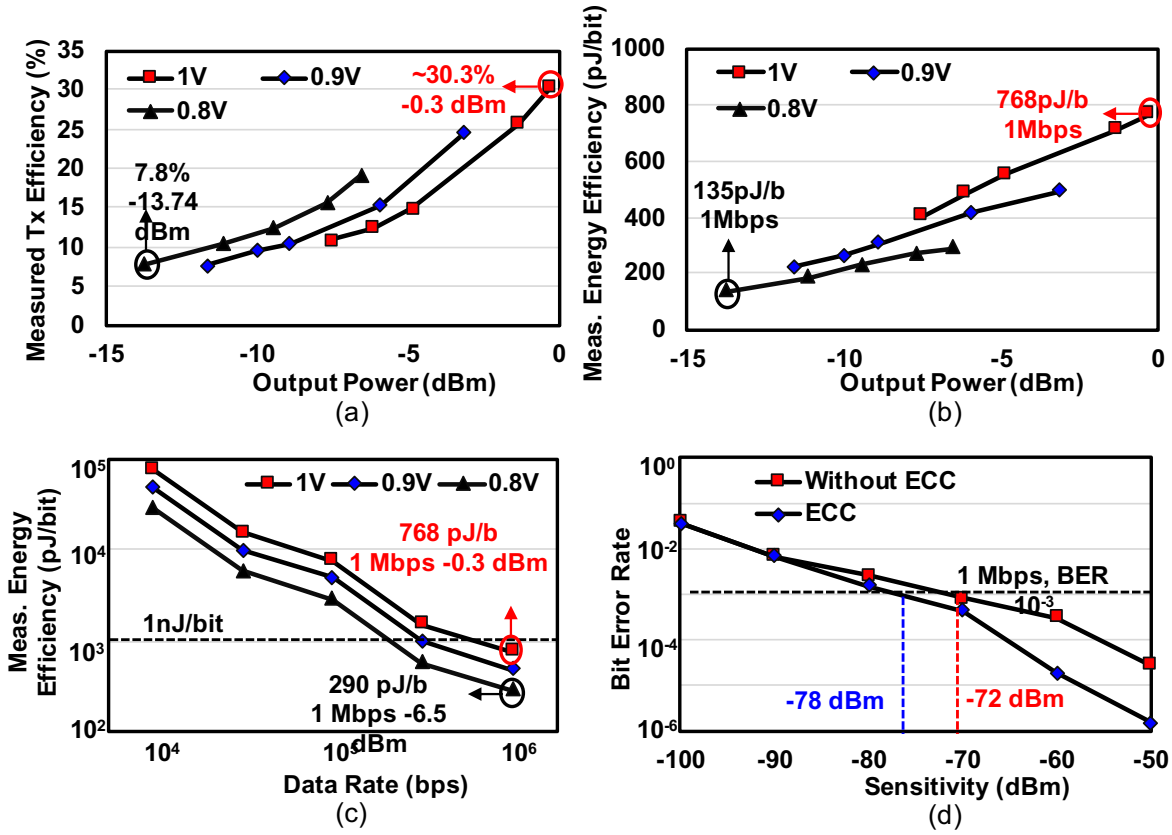


Figure 4.24: Measured transceiver energy performance.

and network sizes and the system autonomously determines the optimal PD to minimize energy, latency or EDP. The online adaptation allows the system to learn and choose the CTRL parameters optimally. We test across various conditions of path-loss, number of edge nodes (i.e., available bandwidth) and obtain a 2.44x (1.47x) improvement in average energy (latency) for a BER of 10-3 compared to the baseline cases while running a modified AlexNet that maps to the SoC (Fig. 4.28).

The proposed system is one of the first prototypes to address computation and communication trade-off with full SoC solution. We have benchmarked our system with state-of-art designs and show competitive figures-of-merit (Fig. 4.29). The design presents a vertically integrated SoC featuring the first real-time NN based adaptation for computation, communication and their trade-offs in energy constrained systems.

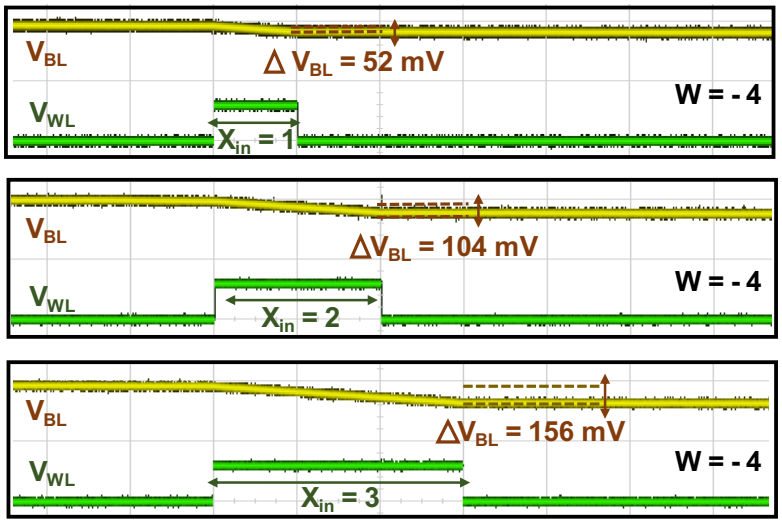


Figure 4.25: Oscilloscope capture of bitline discharge of CUIM module.

4.6 Conclusions

This chapter presents a 65nm wireless image processing SoC for real-time computation-communication trade-off on resource-constrained edge devices. The test-chip includes (1) an all-digital, near-memory, reconfigurable and programmable neural-network (NN) based

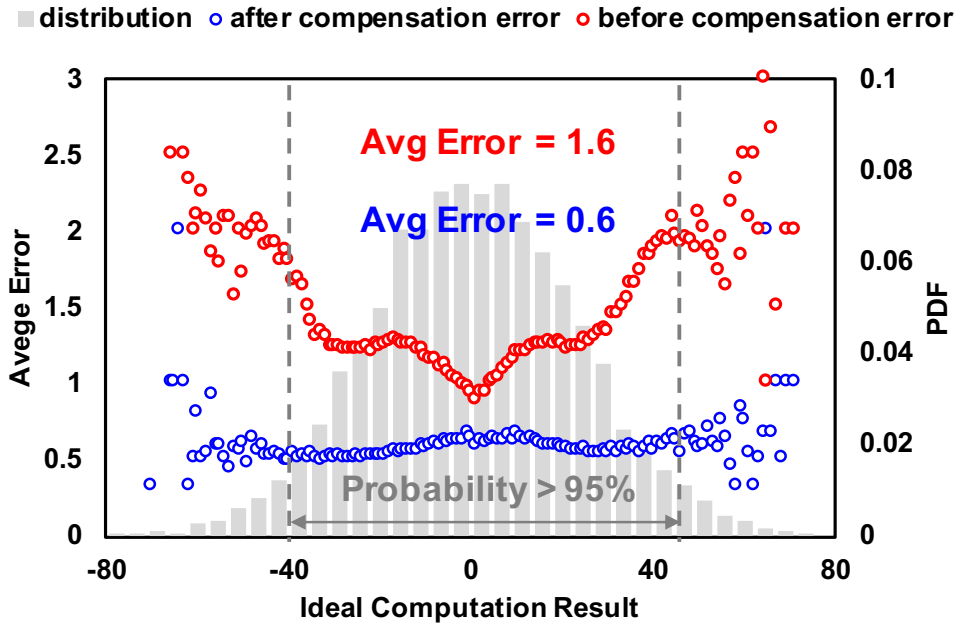


Figure 4.26: Measured CUIM module non-linearities.

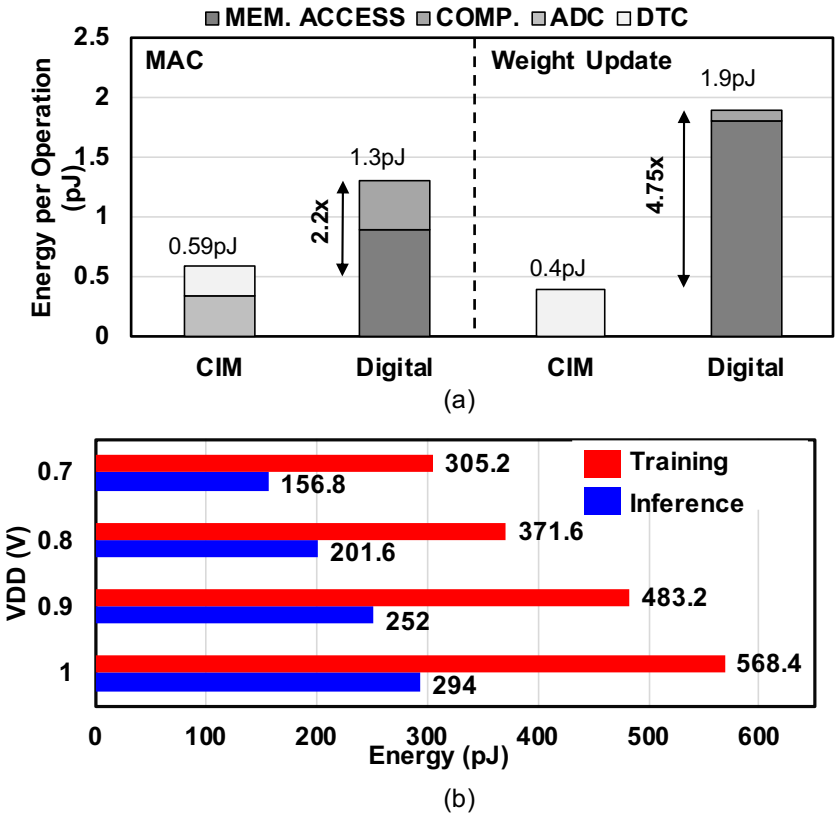


Figure 4.27: Measured CUIM energy efficiency.

systolic image processor at 1.05TOPS/W (peak), (2) a digitally-adaptive RF-DAC based transceiver with Tx energy-efficiency of 768pJ/b and (3) a mixed-signal, time-based, actor-critic neuro-controller with compute-in-memory (CIM) and in-place weight updates that provides online learning and adaptation at 0.59pJ/MAC for efficiently controlling the computation, communication blocks separately as well as jointly.

4.7 Discussions

4.7.1 On-chip System

Compared with the video surveillance platform discussed in previous chapter, proposed SOC has advantages with respect to power, area, real-time transmission reconfigurability, control efficiency and so on. One major reason is that the on-chip system design has shortened the data flow hierarchies through customized circuit design as discussed in proposed

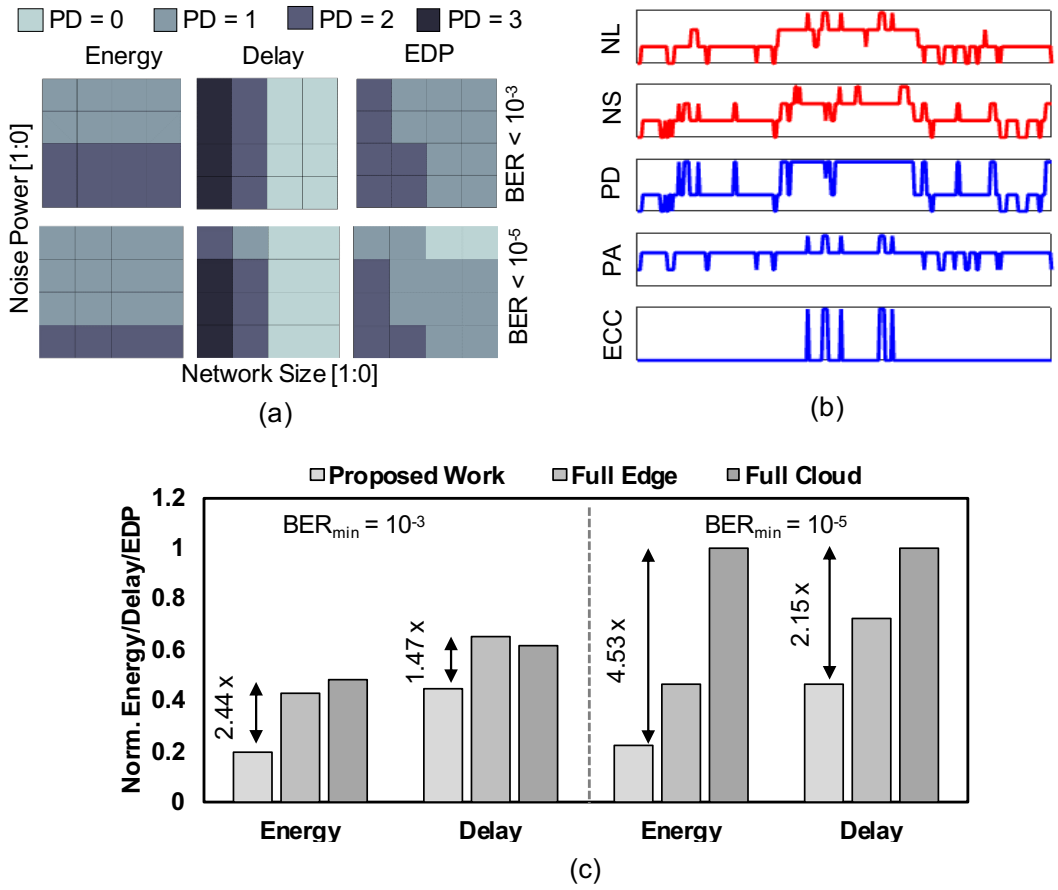


Figure 4.28: System measurements compared with baseline designs.

EI design methodology in Chapter I. In particular for the CIM controller, it utilizes various data-encoding scheme with according circuit techniques (time pulses on WL, charge accumulation on BL/BLB, weight update with pulse modulation and so on). The data movement is minimized and energy is maximally preserved. Further, on-chip design/implementation has greatly improved real-time programmability thus reducing control overhead. In particular, on-chip TRx design has made transmission scheme adjustment delay minimal. In realistic platform, such a implementation can maximally improve its capability to adapt to environment.

		This Work	Machine Learning Accelerators			Adaptive Transceivers		System-on-Chip (SoC)
			Cao ISSCC19	Dongjoo ISSCC17	Suyoung ISSCC17	Arun JSSC13	Arun JSSC13	Tanay ISSCC18
Technology		65nm	65nm	65nm	40nm	65nm	65nm	14nm
SYSTEM	Vcc Range (V)	0.5-1.0	0.4-1.0	0.77-1.1	0.63-0.9	0.7-1.0	/	0.4-1.0
	Area (mm^2)	5mm^2	2mm^2	16mm^2	7.1mm^2	/		6.25
	SRAM (KB)	88	16	280	270			384
	Frequency (MHz)	100-760	<1.5	200	1.9-19.3			0.2-950
	Energy/Frame[1] (uJ)	320	700	420	1960	800	/	1300
COMP	Precision	8b	5b-8b	4b-16b	6-32	/		32
	Peak Energy Efficiency (TOPS/W)	1.05	1.1	1.0-2.1	0.374			0.16
	Norm. Efficiency[2] (TOPS/W)	1.05	1.1-2.9	1.05-2	0.374			0.64
	Peak Energy/Ops (pJ)	0.95	0.22-1.76	0.24-0.5	2.6			/
	Re-configurability	FC, CONV, SP	/	Conv, Recurrent	FC, FFT			CONV
RF	Energy Efficiency (pJ/bit)		/			440	/	/
	PA Efficiency (%)					44.4	25/20	
	Programmability	MOD, ECC, PA Gain				PA, MOD	PA, FIR	
CTRL	Peak Energy Efficiency (TOPS/W)	1.7				/		/
	Peak Energy/Ops (pJ)	0.59						
	Dynamic Control	Actor-Critic Neural Network						PMU
	Optimization	Online Learning						/

[1] Measured and estimated for AlexNet [2] Normalized to 8b operation

Figure 4.29: State-of-art comparison.

4.7.2 Sequential BL Operation Non-linearity

The controller applied in-memory computation scheme. In-memory read, vector multiplication and write and maximally reduce data movement thus improve overall energy efficiency for both inference and learning. However, in the proposed CIM block, the bitline leakage problem demands careful calibration. Bit bitline discharge happens when the rest of cells are not enabled by corresponding word lines, but leakage current drain from precharged bitline to ground through access transistors. This will introduce non-linearity when precharge state is long and/or there are many 0s stored on the bitline side. In this platform, we have applied high-Vt access transistor as well as digital compensation. But intrinsically, it is caused by column-wise sequential bit cell access. The motivation of sequential access is to reduce ADC area by adaptive conversion. It is at the cost of leakage by extending operation time duration on bitline thus introducing leakage non-linearity.

4.7.3 Thermometer-based Encoding

The data-encoding scheme we have adopted for CIM controller is thermometer-encoding. As the bits stored in each latch has equivalent weight compared with binary encoded storage elelment, it is compatible to use pulse modulation to update internal weights. However, we know that thermometer-encoded data lacks density. It consumes $2^n/n$ times more storage for the same information as binary-encoding. For large systems or hgih precision interal weights for neural network, thermometer-encoding should be avoided. It is appropriate for control problem is mainly because the input variable and neural network weight precision are limited.

CHAPTER 5

DISTRIBUTED EI: A UNIFIED COMPUTATIONAL ASIC FOR SWARM ROBOTIC APPLICATIONS

While extensive research efforts have been made to enable intelligence in individual IoT via advanced algorithms, control and system integration for techniques as discussed in Chapter II-IV, scaling these research vectors to multi-agent systems remains a challenging problem. It is critical to investigate collaborative algorithms and hardware architectures that can efficiently collaborate and solve complex problems across multiple agents, such as collaborating robots. This chapter will detail our investigation of a unified computing platform that enables multiple collaborative algorithms on swarm robotics. This chapter is a slightly modified version of "A 65-nm 8-to-3-b 1.0–0.36-V 9.1–1.1-TOPS/W Hybrid-Digital-Mixed-Signal Computing Platform for Accelerating Swarm Robotics" published in IEEE Journal of Solid-State Circuit with the dissertation author as the primary author.

5.1 Introduction

Inspired by the collective intelligence of biological systems, swarm robotics is an emerging area where multiple robots work together to enable complex swarm behaviour. The problem solving capability enabled through simple interactions among the agents enables novel applications [102, 103, 104, 105, 106]. In swarm robotics multiple small and distributed robots co-ordinate and gather data to enable intelligent decision-making as a group. These have been used in applications such as exploration, reconnaissance and disaster relief [107]. The fact that distributed and swarm robotics are resilient to component-level failures further motivates the use of swarms. In swarm robotics, multiple robots often co-ordinate in real-time to solve diverse problems such as pattern-formation, cooperative reinforcement learning (RL), path-planning etc. Some of these algorithms use learning-based methods

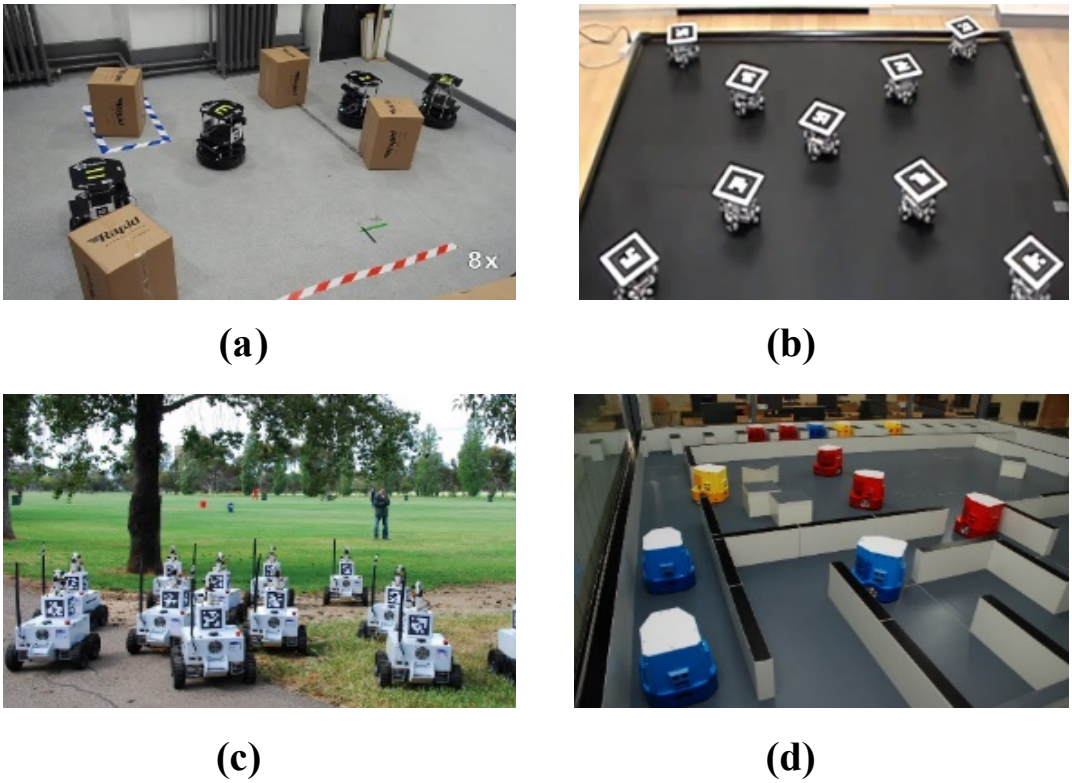


Figure 5.1: Swarm algorithms that can successfully accomplish (a) collaborative path-planning (b) pattern formation; (c) multi-agent patrolling; (d) multi-agent predator-prey.

and have gained increasing importance with the success of deep neural networks and neuromorphic computing. Although certain swarm algorithms rely on real-time learning (e.g., cooperative RL) representing a model-free approach, many powerful algorithms that have been developed over the past two decades (e.g., pattern formation) rely on a mathematical structure and represent a more traditional physical-model-based approach. The next generation of swarm hardware needs to support both of these approaches; and hence, it is important to identify the common computational kernels that need to be supported in hardware. However, hardware designs that can support computation in swarms is computationally challenging; especially from an energy-perspective. This is discussed in [108]: main processor in a coin-size swarm robot consumes $4\times$ energy than a micro-controller, and this energy is comparable (more than 80%) to motors and camera based sensors [108]. As swarm robots are expected to enable so-called "intelligence" in reduced form fac-

tors, energy-efficient hardware design continues to be an active area of research. In this chapter, we identify the commonalities and shared compute primitives across a variety of model-based and model-free swarm algorithms and present a unified, fully-programmable, energy-efficient and scalable platform capable of real-time swarm intelligence. Although we demonstrate how to support some sample algorithms here, the design principles are scalable and can be applied to larger swarms enabling more advanced algorithms.

To enable a unified energy-efficient computing platform for swarm robotics, we demonstrate a hybrid mixed-signal and digital design. In [109], we demonstrated a purely time-based mixed-signal neural network for reinforcement learning on edge devices. However, a purely mixed-signal solution shows superior energy-efficiency for low bits of resolution. As the number of bits on the data-path increases, mixed-signal solutions tend to be less efficient than purely digital counterparts. In swarm robotics, the size of the swarm determines the size of vectors that need to be computed and hence the bit-width required for high-accuracy also scales with the swarm size. Hence, mixed-signal solutions are efficient for small swarms while digital solutions tend to out-perform in larger swarms. To enable such scalability, we demonstrate a hybrid digital-mixed-signal solution where a time-domain mixed-signal kernel computes on 3b-5b data. A digital wrapper around the mixed-signal kernel further scales the computing platform to 6b-8b. This allows high energy-efficiency for low-precision along with the excellent energy-scalability of digital computing for larger bit-widths.

The test-chip has been fabricated in a 65nm CMOS process. We demonstrate 9.1TOPS/W peak energy-efficiency at 3b of resolution. The energy-efficiency decreases to 1.1TOPS/W for 8b resolution. The test-chip interfaces with a raspberry-Pi platform consisting of integrated sensors (inertial sensors and ultrasonic distance sensors) and LoRa (Long Range) radios for decentralized, peer-to-peer communication among mobile robotic vehicles in a swarm. The rest of the chapter is divided as follows. Section II provides an overview of the swarm algorithms. The next two sections describe the scalability of the computing platform

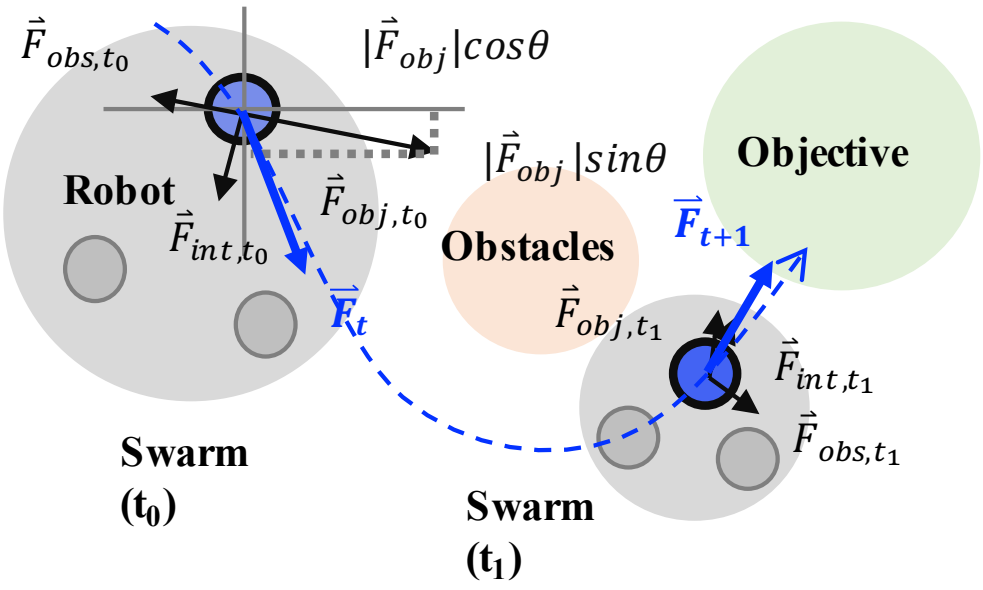


Figure 5.2: Schematic map showing APF-based path-planning and formation.

with the swarm size and the hybrid-digital-mixed-signal design. The system overview is described in Section V and measurement results are shown in Section VI. Finally outlook of potential future works are discussed in Section VII and conclusions are drawn in Section VIII.

5.2 An Overview of Swarm Algorithms

Swarm algorithms can be broadly classified into two categories: the ones based on physical and mathematical models and the ones based on learning. In the following subsections, we provide an overview of the types of algorithms that are supported by the common unified platform.

5.2.1 Algorithms Based on Physical Models

Over the past decades, there have been significant development in swarm control algorithms inspired by physical and mathematical models. Among these mathematical models, Artificial Potential Field (APF) is a popular and practically useful computational approach.

In APF, we assume that the robots and the objects (goal, obstacles, teammates and etc.) are similar to 'electrical charge' that produce artificial attractive and repulsive potential fields whose potential functions are to be leveraged by the system designer for optimal robotic control and system performance. By aggregating the potential fields (i.e., forces), the motion vector can be obtained at each evaluation step. In general, APF algorithm has the following format [110, 111, 112, 113, 114]:

$$m_{\text{i}} \frac{d\vec{v}_{\text{i}}}{dt} = \vec{F}_{\text{pro,i}} + \vec{F}_{\text{int,i}} + \vec{F}_{\text{esp,i}} + \vec{F}_{\text{est,i}}; \tag{5.1}$$

This is based on Newton's second law to describe the i_{th} robot's velocity v_i change determined by propulsion $\vec{F_{\text{pro,i}}}$, interaction $\vec{F_{\text{int,i}}}$, objective escape $\vec{F_{\text{esp,i}}}$ and stochastic forces $\vec{F_{\text{est,i}}}$ and mass m_i. By properly choosing the potential function that generates each term, we are able to design a cooperative control algorithm that can implement applications such as collaborative path-planning, co-coordinated formation and etc.. A typical example is shown in Fig. 5.2.

For example, for path-planning applications as demonstrated in Fig.5.1.(a), the positional information of objectives and obstacles are required in determining motion vectors. In this design, we consider standard parabolic potential U_{obj} for the object and an exponential potential barrier for the obstacles U_{obs} from [110]:

$$U_{\text{obj}}(\vec{r}) = k_{\text{obj}} dis(\vec{r}, \vec{r}_{\text{obj}})^2; \tag{5.2}$$

$$U_{\text{obs}}(\vec{r}) = k_{\text{obs}} dis(\vec{r}, \vec{r}_{\text{obs}})^{-1}; \tag{5.3}$$

where $\vec{r}_{\text{obj}}$ and $\vec{r}_{\text{obs}}$ are positions of the objective and obstacles respectively. The force

vectors created by these potential functions in the 2-D plane are of the form:

$$\vec{F_{\text{pro}}} = -k_{\text{f}}\nabla_{\text{i}}(U_{\text{obj}}(\vec{r}) + \sum_{m=1}^{M} U_{\text{obs}}(\vec{r})); \tag{5.4}$$

$$F_{\text{pro,x}} = \alpha|\vec{r} - \vec{r}_{\text{obj}}|cos\theta_{\text{obj}} + \sum_{m=1}^{M} \beta_{\text{m}}|\vec{r} - \vec{r}_{\text{obs,m}}|^{\text{-2}}cos\theta_{\text{obs,m}}; \tag{5.5}$$

$$F_{\text{pro,y}} = \alpha|\vec{r} - \vec{r}_{\text{obj}}|sin\theta_{\text{obj}} + \sum_{m=1}^{M} \beta_{\text{m}}|\vec{r} - \vec{r}_{\text{obs,m}}|^{\text{-2}}sin\theta_{\text{obs,m}}; \tag{5.6}$$

For formation applications as demonstrated in Fig.5.1.(b), the potential function uses a logarithm-cosine-hyperbolic function:

$$U_{\text{int}}(\vec{r}) = \beta ln(cosh|\vec{r} - \vec{R}|); \tag{5.7}$$

where $\vec{r}$ is the interaction vector while $\vec{R}$ is the target vector. Enabling each interaction with a dedicated target vector allows fine tuning of the shape of the formation. The resulting force equations in the 2-D plane can be expressed as:

$$\vec{F_{\text{int}}} = -\nabla_{\text{i}}(U_{\text{int}}(\vec{r}_{\text{i}}, \vec{r}_{\text{i}}); \tag{5.8}$$

$$F_{\text{int,x}} = \sum_{m=1}^{M} \alpha_{\text{i}}[tanh(|\vec{r}_{\text{j}} - \vec{R}_{\text{j}}|)cos\theta_{\text{j}}]; \tag{5.9}$$

$$F_{\text{int,y}} = \sum_{m=1}^{M} \alpha_{\text{i}}[tanh(|\vec{r}_{\text{j}} - \vec{R}_{\text{j}}|)sin\theta_{\text{j}}]; \tag{5.10}$$

To solve swarm problems we need to compute equation eq. (5.1) with the correct parametric representations of the functions and parameters as obtained from eqs. (5.4) to (5.6) and eqs. (5.8) to (5.10). These parameters are obtained from system level simulations before

deployment.

5.2.2 Learning-Based Algorithms

With the rapid development of hardware systems to support machine learning and artificial intelligence [115, 116, 109, 117], advanced learning-based techniques are becoming popular for applications such as multi-robot predator-prey and multi-agent patrolling as shown in Fig. 5.1.(c-d), Learning-based algorithms have now become competitive in a variety of problems where pre-defined models may not exist or may be incomplete. The motivation for the learning-based approach is to allow each robot to learn continuously without human intervention and establish a control model with real-world knowledge. Among all the approaches, reinforcement learning (RL) based cooperative Q learning [103, 118, 119, 120] algorithm has shown great promise.

Single-agent Q learning[121, 122] is based on the iterative update of the Q value, as a robot navigates through a series of (state, action, reward) tuples. This iterative scheme is derived from the Bellman equation [123] for optimal control. The iterative algorithm can be summarized as:

$$\begin{aligned} Q_{\mathrm{t+1}}(S_{\mathrm{t}}, A_{\mathrm{t}}) = Q_{\mathrm{t}}(S_{\mathrm{t}}, A_{\mathrm{t}}) + \\ \alpha(R_{\mathrm{t}} + \gamma max Q_{\mathrm{t}}(S_{\mathrm{t+1}}, A_{\mathrm{t}}) - Q_{\mathrm{t}}(S_{\mathrm{t}}, A_{\mathrm{t}})); \end{aligned} \tag{5.11}$$

$$R_{\mathrm{t}} = f(S_{\mathrm{t}}); \tag{5.12}$$

where γ and α are discount factor and learning rate to aggregate the distant rewards and update Q tables respectively. By taking a series of actions A (moving forward, backward etc.) in the state space S (positions, obstacle vectors and etc.), the robot calculates the reward for each action and updates the Q-table, thus creating a robust functional mapping from the state space to the action space. The reward is based on a single robot's current state. A hardware implementation of Q-learning for autonomous navigation has been presented

in [109, 124] and interested readers are pointed to the references for more details.

In cooperative Q learning, global , instead of local, states and rewards are utilized to facilitate multi-agent collaboration. As opposed to the baseline Q-learning where a single-agent's local state is used, in a swarm, the local states are broadcasted to all the teammates. This forms a global state which incorporates the knowledge of all teammates. The Q-value of the swarm is now evaluated as:

$$\begin{gathered} Q_{\text{t+1}}(S_{\text{t,global}}, A_{\text{t}}) = Q_{\text{t}}(S_{\text{t,global}}, A_{\text{t}}) + \\ \alpha(R_{\text{t,global}} + \gamma max Q_{\text{t}}(S_{\text{t+1,global}}, A_{\text{t}}) - Q_{\text{t}}(S_{\text{t+1,global}}, A_{\text{t}})); \end{gathered} \tag{5.13}$$

$$S_{\text{t,global}} = [S_{\text{t,1}}, S_{\text{t,2}} ... S_{\text{t,N}}]; \tag{5.14}$$

In a manner described in [124], each robot will now take an action based on the best Q value of current global state. A global reward is evaluated based on the team's performance, for example, whether one of the targets has been reached by one of the team members. It is worth noting that we incorporate the task completion time as a reward function, as it improves the swarm's performance and facilitates convergence by encouraging all robots to take continuous actions.

$$R_{\text{t}} = g(S_{\text{t,global}}, t); \tag{5.15}$$

When the environment is complex and the swarm size is large, the global state can also be significantly large. It is difficult to store all the Q values in a table, especially in memory-constrained design. Therefore, the Q value is typically approximated as a neural network output. The states ($S_{\text{t,global}}$), (sensor values, current positions etc.) act as inputs to the neural network. Then every neural-network propagates the states through embedded neural network and produces Q- values of each action. A hard-max function at the end of the neural network establishes the best action to be taken. We use ϵ-greedy as means to perform exploration. Details of co-operative Q-learning for multi-robot action is a rich and evolv-

ing area of algorithmic research. For more details on co-operative Q-learning, interested readers are directed to [103].

5.3 A Common Computing Platform

As we mentioned earlier, future computing platforms that can support swarm algorithms, need to support both mathematical algorithms as well as learning-based algorithms. Interestingly, we observe that both these two algorithms have a basic mathematical structure. As computational problems, they both feature:

1. **Linear Processing Unit:** Both types of algorithms work on vectors and matrices and hence linear processing is a critical components of computation. In APF based algorithms, linear operations are performed on trigonometric transformations of motion vectors (eq. (5.1)). In neural networks, the linear units allow the synaptic weights to be summed up at the input of a neuron. Fundamentally, the computational platform needs to support multiplications and additions (through multiply-and-accumulate, MAC units).

2. **Non-linear Processing Unit:** Apart from linear vector processing, both algorithms require non-linear transformations. In APF algorithms, these transformations are mostly trigonometric (eqs. (5.4) to (5.6) and (5.8) to (5.10) whereas in neural networks these transformations are the activation functions (sigmoid, ReLU etc.). In APF, the linear processing is done on non-linearly transformed motion and position vectors; hence we perform non-linear processing followed by linear processing. On the other hand, in neural networks, we perform MACs first, followed by non-linear activation functions.

Since linear/non-linear operations are the major workloads in robotic algorithms, this unified compute platform is designed to provide a unified solution to accelerate both types

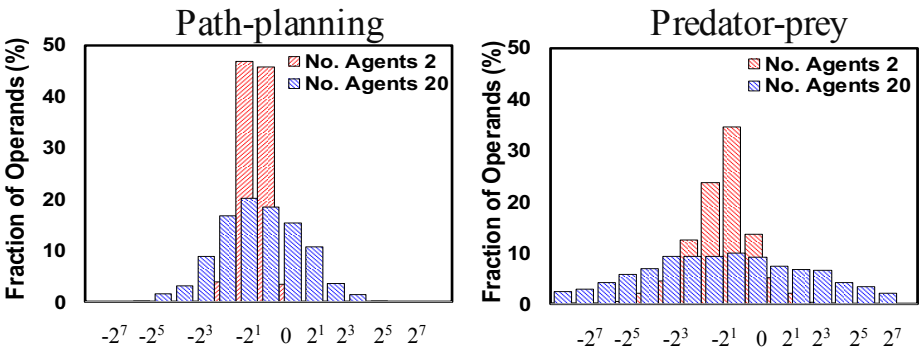
of computation. With a dedicated Non-linear Processing Unit and a Linear Processing Unit, we achieve high energy-efficiency as will be described in the later sections.

The order of linear and non-linear processing are different in the two algorithms but in a memory-centric system, this amounts to simply changing the order of instructions to support both classes of algorithms. This shows that a unified computing platform comprising of (1) a linear processing unit, (2) a programmable non-linear processing unit, (3) a data-cache and (4) an instruction cache will be able to support both the model-based and learning-based algorithms. In the proposed ASIC, we demonstrate support for both types of algorithms with a non-linear processing unit which is composed of a look-up table based piece-wise approximation of the non-linear function. The linear processing unit is composed of a MAC array and data-cache and instruction-cache with standard 6T SRAM cells.

5.4 Scalability with Swarm Size

The number of agents in a swarm, also called the swarm size, is a major design parameter for providing optimal performance and robustness at minimal system cost. For example, in disaster relief, to ensure the largest area coverage and fastest convergence rate a relatively large number of agents is often preferred. However, for indoor exploration, a small group of robots is likely to be sufficient given the reduced problem complexity, and increased environmental clutter. As a consequence, future computing platforms that can support multiple swarm algorithms also need to be able to support multiple swarm sizes. To prevent over-design, the computing platforms need to perform at optimal energy-efficiency for a large scale of swarm sizes.

To better understand the computation requirement for varying swarm sizes, we analyze both the model-based and learning-based algorithms as a function of the swarm size. In model-based APF swarm control, the mathematical structure of the problem follows a



(a) **(b)**

Swarm size	Algorithms			
	Path-planning	Formation	Predator-prey	Cooperative exploration
2	3	3	5	4
5	4	4	7	4
10	5	5	7	5
15	5	6	8	5
20	6	7	8	6

(c)

Figure 5.3: Algorithmic simulations demonstrate how the required bit-precision scales with the swarm size for two template problems: (a) collaborative path-planning and (b) multi-agent predator-prey. The number of bits required to accurately compute different template algorithms for varying swarm sizes is shown in (c).

general form:

$$\vec{F} = \sum_{m=1}^{M} NL_{\mathrm{m}}(\vec{d}_{\mathrm{m}}); \tag{5.16}$$

where $\vec{F}$, NL_{m} and $\vec{d}_{\mathrm{m}}$ represent the aggregated potential field force vector, the $\mathrm{m_{th}}$ nonlinear function and $\mathrm{m_{th}}$ distance vector respectively, while M is the total number of vectors. On the other hand, for learning-based cooperative RL algorithms, as the Q table is approximated by the neural network, the general computation paradigm is the same as computing each neuron's output:

$$y_{\mathrm{j}} = a(\sum_{i=1}^{N} w_{\mathrm{i,j}}(x_{\mathrm{i}})); \tag{5.17}$$

here x, w, y and a are the inputs, weights, neuron outputs and nonlinear activation functions

respectively while N is the number of pre-synaptic neurons. It is easy to understand that M will scale with swarm size, especially in applications such as pattern formation. Similarly, N is determined by the dimension of the global states of the system which scales with the swarm size. As a result, a larger swarm will require a wider range of operands, thus requiring a higher bit-precision to correctly process APF algorithms as well as co-operative RL. Fig. 5.3(a-b) demonstrates simulation results of representing the required range of the operands for different swarm sizes in both physical model-based (coordinated path-planing) and well as learning-based (multiple predator-prey) template algorithms. We note that as the swarm size increases, the bit-precision required to correctly compute also increases. The simulation results can be summarized in the Fig.5.3.(c) where the template algorithms that can be supported require a bit-width of 3b to a maximum of 8b. In these applications, the sensor-data are assumed to have a bit width of 8b or less and obstacle-avoidance is performed using ultra-sonic sensors.

5.5 Hybrid Digital-Mixed-Signal Computing

The advantage of using analog and mixed-signal design principles for energy-efficient computing have been demonstrated in [109, 124, 125]. More recently, there has been increasing interest in time-based mixed-signal computing. Here information is represented in phase or frequency domain and hence the effective number of bits is not limited by the voltage-scalability of the design. However, since the data is processed in time-domain, the system throughput is lower than corresponding digital systems. For many problems of practical interest, in particular for control and robotics on small form factors where the data-processing speed is relatively low, this is a favorable trade-off. It has been demonstrated successfully in RL problems [109, 124] as well as in convolutional neural networks [126], decoders [127] and pipelines circuits [128]. In spite of its superior energy-efficiency at low bit-widths (typically less than 5b or 6b depending on the process), it is well understood that as the bit-precision scales to high values, the energy-efficiency of digital circuits take-

over. Hence, for the problem at hand, where an increasing swarm size should be supported with a higher bit-width, an ideal system should scale seamlessly between a mixed-signal (time-based) to a digital design such that a high energy-efficiency is obtained as the system specifications scale.

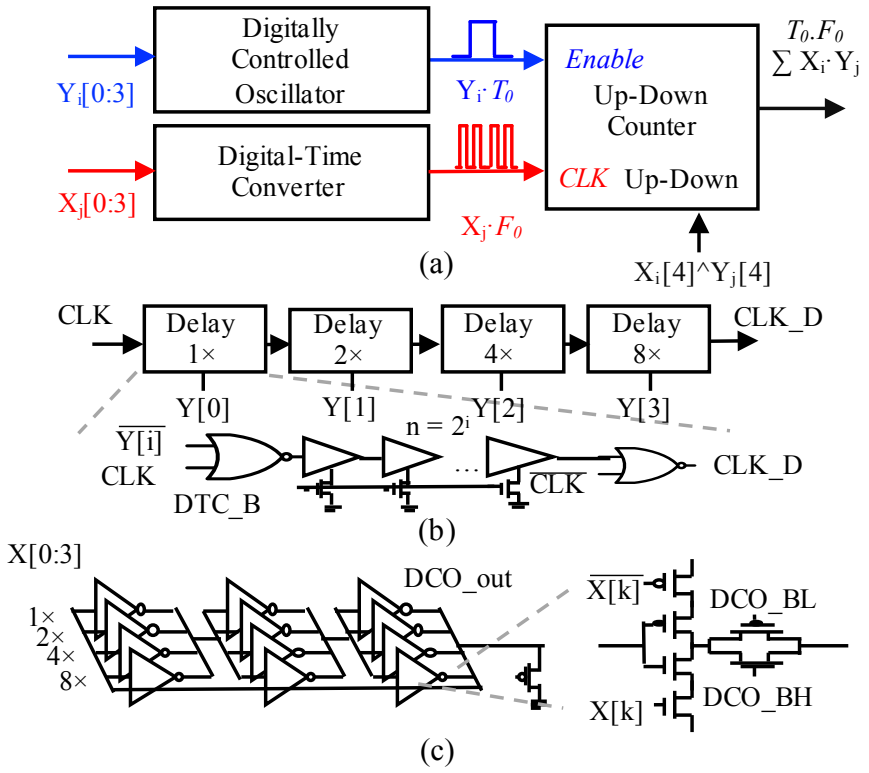
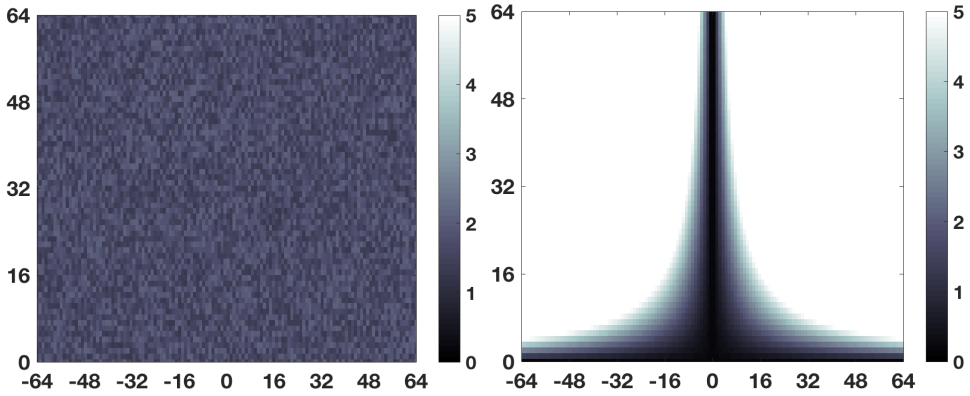


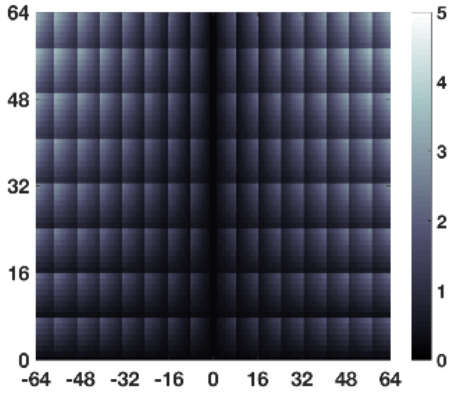
Figure 5.4: Circuit schematic illustrating (a) the time-domain-mixed-signal MAC circuit (b) the digital-to-pulse-converter (DPC) (c) the digitally-controlled-oscillator (DCO).

5.5.1 Time-Domain Multiplication and Accumulation

The details of time-domain multiplication and accumulation have been described in [109] and will be summarized here for completeness. Fig. 5.4.(a-c) illustrates the time-based multiply-and-accumulate (MAC) circuit. The time-based circuit operates on 5b data representing both positive and negative numbers. It has a pulse input (T_p) used as the "Enable" signal to an up-down counter. Signed operation is handled by XOR operation of



(a) (b)



(c)

No. Bits	TD-MS		HDMS	
	Ave	Worst	Ave	Worst
3	0.10	0.49	0.19	0.52
4	0.14	0.56	0.16	0.61
5	0.28	0.72	0.29	0.74
6	0.64	1.74	0.69	0.94
7	2.21	3.86	0.70	1.02
8	5.82	9.32	0.69	1.27

(d)

Figure 5.5: Energy map vs. operand range in pJ for (a) digital (b) TDMS and (c) HDMS MAC implementations. (d) The energy/MAC (normalized to a digital implementation) for TDMS and HDMS implementations. We see that HDMS out-performs TDMS (average and worst cases) and digital (average case) for large swarm sizes.

sign bit as indicated in Fig. 5.4.(a). One of the operands (X[0:4]) is encoded in the pulse-width of T_p using a digital-to-pulse-converter (DPC) with X[4] as sign bit. For the i^{th} input X_i, we obtain:

$$T_{p_i} = X_i * T_0 \tag{5.18}$$

where T_0 is the unit time-constant for the DPC. The other input (Y[0:4]) is encoded in the signed magnitude format and controls a digitally-controlled oscillator (DCO). Y[4] represents the sign bit and Y[0:3] represents the magnitude of the second operand. The

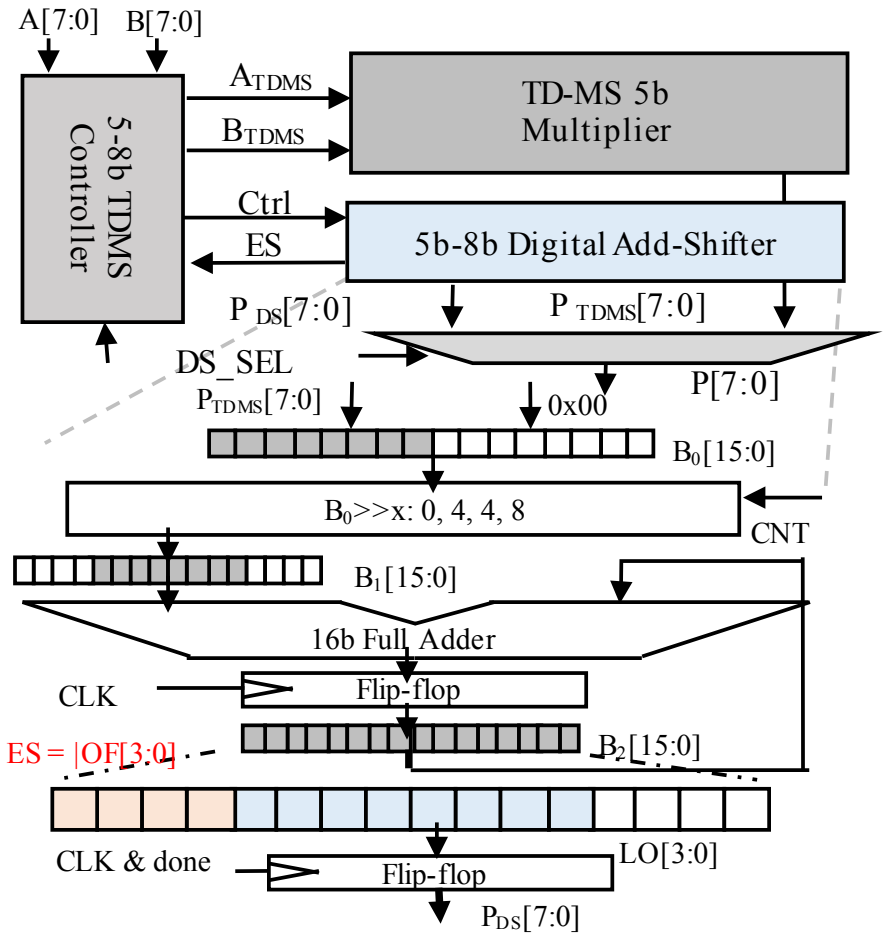


Figure 5.6: Circuit schematic of the HDMS circuit illustrating the 5b TDMS kernel and the digital peripherals to enable efficient scaling to 8b.

3-stage DCO converts the digital value to a frequency proportional to Y[0:3]. Each stage of the DCO consists of a bank of parallel binary-sized inverters controlled by the digital value (Y[0:3]) as shown in Fig. 5.4.(b-c). The frequency of the DCO for the j^{th} word (Y_j), is F_j and ignoring second-order effects such as non-linearity, is given by:

$$F_j = Y_j * F_0 \tag{5.19}$$

where F_0 is the unit frequency of the DCO corresponding to a $code\ 1$ when $W = 00001$. The clock to the counter is driven by the DCO, and the enable signal is controlled by the

pulse width (T_{p_i}). Hence, the counter output is given by:

$$DT_{ij} = T_{p_i} * F_j = X_i * Y_j * F_0 * T_0 \tag{5.20}$$

From. eq. (5.20) we can observe that the counter output is proportional to the product of the two operands. As shown in Fig. 5.4.(a), polarity of MAC is taken care of through up-down knob of counter controlled by XOR of X[4] and Y[4]. The constants, F_0 and T_0 represent the overall system throughput and designed to maintain correct functionality amidst non-linearities. The scalability of this design to a large number of vector-parameters has been discussed in [124, 125].

5.5.2 Hybrid-Digital-Mixed-Signal Computing Platform

It is worth noting that the time-domain MAC shows high energy-efficiency for low bit-widths only. Fig. 5.5.(a-b) illustrate the simulation results of a 65nm CMOS GP process and it reveals that the energy consumed for a MAC operation scales faster than a digital system. This can be intuitively understood from the fact that the number of switching events (in the worst-case) for a time-domain phase-frequency based design scales as 2^N for N-bit operands. This results in an interesting artifact, where important computation (where operands have higher magnitudes) consumes more energy than less important computation (where operands have lower magnitudes). The 2D energy-bar shown in Fig.5.5 illustrates how a time-domain system shows high energy-efficiency for bit-width less than 5, but it increases dramatically as the bit-width increases. To maintain high efficiency across the entire operating range, we proposes a hybrid-digital-mixed-signal (HDMS) MAC kernel as shown in Fig.5.6.

The HDMS MAC kernel consists of a conventional time-domain mixed-signal (TD-MS) multiplier, a 5b-8b digital add and shifter and a 5-8b TD-MS controller. For bit precisions less than 5b, the circuit operated completely in the time-domain. The idea is to compute an 8 bit multiplication via shift-and-add. At the core, we have an energy-efficient

time-domain 5b multiplier. Around that, we have peripheral circuits (add-shifter and controller) to reconfigure the multiplier to higher bit precision, as needed by allowing seamless shift and add operations. The 5b-8b digital add-shifter circuit diagram is shown in Fig.5.6. The figure illustrates the HDMS circuit: a shifter shifts the TD-MS products by 0, 2, 4 and 8 bit each time and accumulates through time with a 16b full adder. The computation starts from the most significant bit and proceeds to the least significant bit. This helps us to save unnecessary switching by stopping the computation as soon as any overflow is detected through the embedded overflow detection. By driving a digital select signal (DS_SEL) active, the 5-8b TD-MS controller splits 8 bit input operands A and B into 4 bit components, passes them to TD-MS multiplier in pairs, and controls the add-shifter to produce high precision output. With the proposed kernel, we are able to preserve the energy-efficiency of the time-domain computation for lower bit-precision, while leveraging the efficiency of digital computation for higher bits of precision. The energy map of HDMS is demonstrated in Fig.5.5.(c) and TDMS/HDMS energy normalized to digital circuit with same bit-precision is shown in table in Fig.5.5.(d). It should also be noted that the proposed scheme is scalable to handle more than 8b operations.

We should also note that, although HDMS requires additional clock cycles ($4\times$) than TD-MS, it still shows higher throughput; owing to the fact that HDMS avoids long clock periods, typical of 8b TDMS ($16\times$). With both energy and throughput advantages, the major trade-off is the additional area required for the digital peripherals. However, it should be noted that HDMS achieves lower throughput than high-speed digital. In the current application, the throughput that we achieve is more than sufficient to support the data rate for the sensors and actuators.

5.6 System Overview

The system architecture of proposed computation platform is illustrated in Fig.5.7. As mentioned in section IV, we have noted that APF and cooperative RL are essentially com-

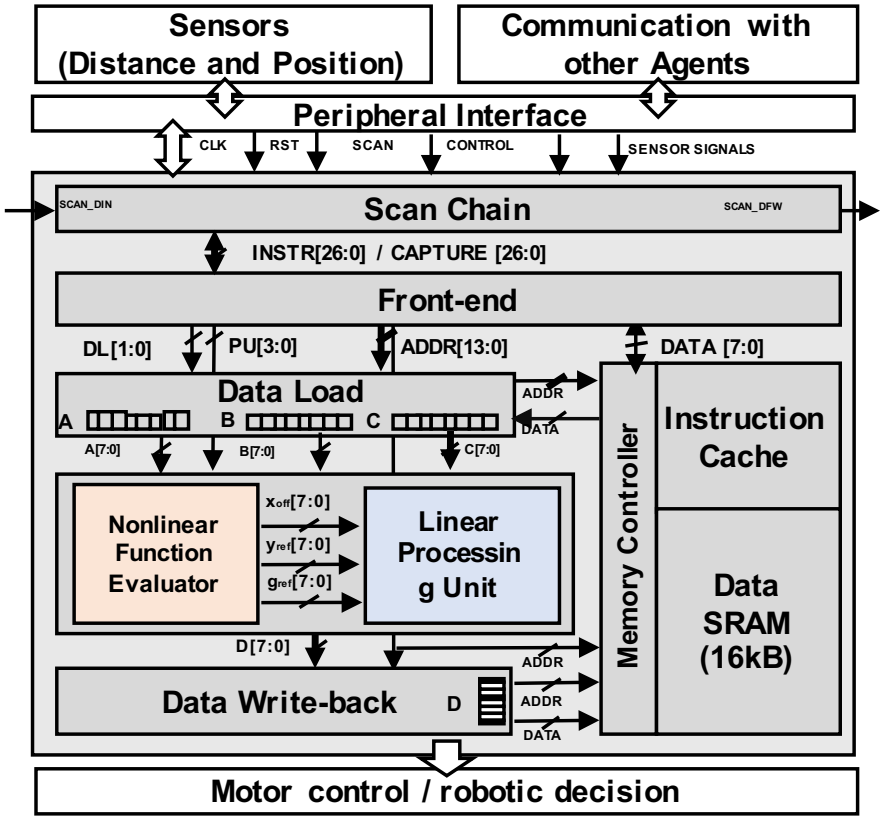
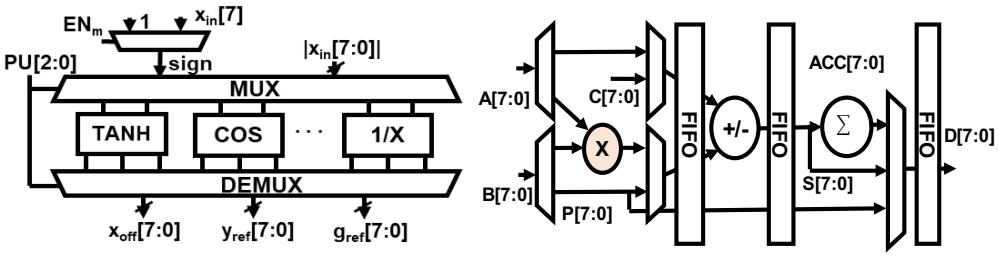


Figure 5.7: Overall system architecture of the unified computing platform.

binations of nonlinear evaluations and linear operations. This has inspired us to design a dedicated accelerator for nonlinear and linear computations, which are called the Nonlinear Function Evaluator (NFE) and the Linear Processing Unit (LPU) respectively. NFE implements the non-linear function using piece-wise linear approximation of the nonlinear functions. We embed a number of widely used nonlinear functions in the NFE. By choosing the function to evaluate and providing the input parameter, NFE generates an offset (x_{off}), a reference gradient (g_{ref}) and a reference offset (y_{ref}) in one clock cycle. The corresponding evaluation result is generated by multiplication/addition of x_{off}, g_{ref}, y_{ref} in the LPU. The number of clock cycle depends on the bit-precision selected. We observe that many of the required functions show symmetry or periodicity, and we take advantage of that to implement a mapping mechanism to reduce the number of comparisons and computations. This saves active die-area as well as computational energy. The reference parameters are stored

in a look-up-table (LUT). By storing only the important parameters, determined from the range of the inputs and by interpolating in the LPU, NFE is achieving target accuracy for the entire range of the data. As opposed to using a LUT for the complete range of inputs, the proposed design allows a compact implementation with a reduced memory foot-print. On the other hand, the LPU supports all the linear operations (addition and multiplication). Most operations are implemented in the digital domain except for multiplication and accumulation (MAC). Circuit and control details of NFE and LPU are illustrated in Fig.5.8.(a-d).



(a)

PU	FUNC	EN	seg	#comp
000	Tanh	1	10	5
001	Cosine	0	12	6
010	Sqrt	1	4	4
011	step	1	2	1
100	ReLU	1	2	1
101	X^2	0	12	6
110	1/X	1	10	5

(b)

(c)

INS	OP	#Bytes	#Clk
000	A+B	2	1
001	A-B	2	1
010	AB+C	2 / 3	2 / 6
011	∑ A	N	N+1
100	∑ AB	N / 2N	N+2/5
101	∑ AB+C	2 / 3N	N+3/6

(d)

Figure 5.8: (a) Circuit schematic and (b) the corresponding control bits for the NFE. (c) Circuit schematic and (d) the instructions for the LPU.

We provide bi-directional local data path between LPU and NFE for computations. Data can move between the LPU and the NFE seamlessly to preserve data-locality.

A 16KB on-chip SRAM is embedded together with an instruction cache, a data loader and write-back controllers. A front-end controller is also provided and the design is full-scan. It should be noted that, either in model-based or learning-based applications, required information storage will scale with the swarm size and the complexity of the environment.

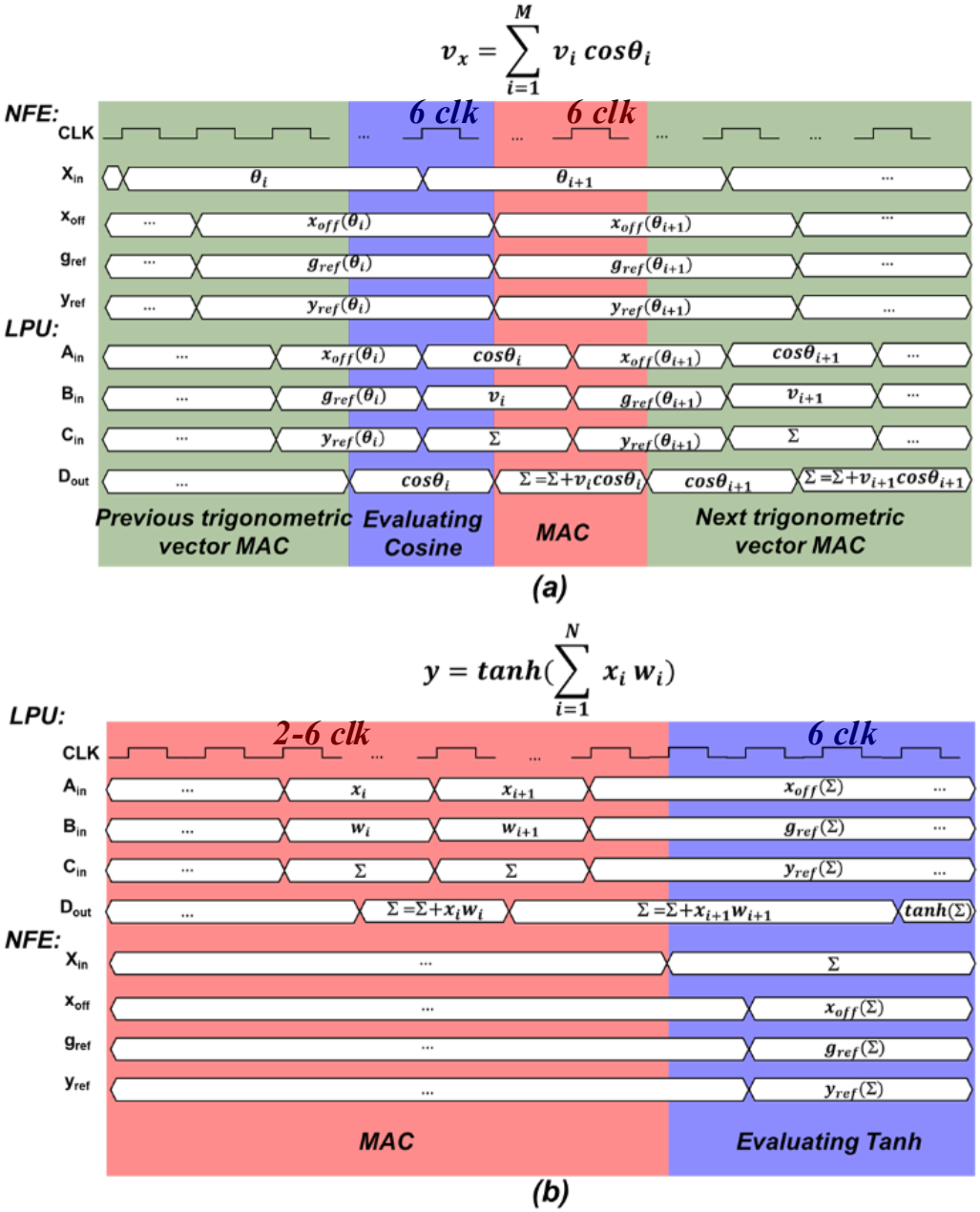
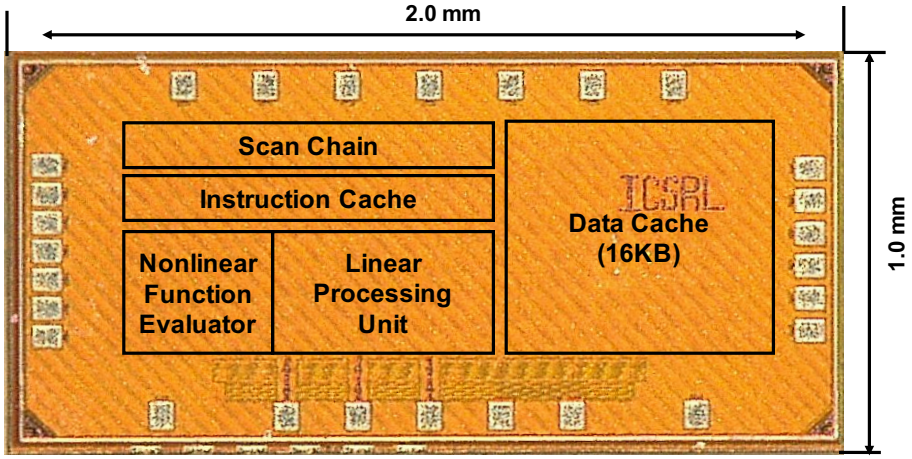


Figure 5.9: Clock diagram for examples in APF (a) and cooperative RL(b).

The current design is a prototype with 16KB of on-chip memory. For more complex "experience maps", off-chip storage is required. This is not supported in the current test-chip. With the embedded computation/storage capability, the chip is able to interface with sensors and communication components for swarm robotics. The sensors and actuators interface through a Raspberry PI, which acts simply as an interface. All the sensors produce



Chip Characteristics	
Technology	**65nm 1P9M CMOS**
Die area	**1mm*2mm**
Testing interface	**QFN package**
Pin Count	**28**

Figure 5.10: Die photo and chip characteristic.

.

digital outputs. Ultra-sonic sensors are used for depth measurements. IMUs are used to estimate position, by integration in the Raspberry PI. In future work, the system may be scaled to enable more complex mapping and localization algorithms. With limited on-chip resources, this test chip is intended to work as a co-processor to support key algorithms and

	This work	[109]	[129]	[130]	[131]	[117]	[126]	[128]
Application	Swarm Learning	Autonomous micro-robotics	CNN Inference	DNN Inference	CNN Inference	CNN Inference	CNN Inference	DTW
Optimization algorithm	Cooperative RL/potential field	Reinforcement Learning	none	none	none	none	none	Time-series Classification
Learning/Training	Online real-time	Online real-time	offline	offline	none	offline	offline	none
Technology	65nm	55nm	180nm	65nm	65nm	65nm	40nm	65nm
Aream	$2mm^2$	$3.4mm^2$	$3.3mm^2$	$16mm^2$	$16mm^2$	$16mm^2$	$0.124mm^2$	$1.67mm^2$
On-die SRAM	16KB	200B	144KB	36KB	490.5KB	181.5KB	/	/
Resolution	5-8b	6b	4b-16b	16b	16b	16b	8/1b	4/10b
Power	0.3-3.4μW	650μW	7.5-300mW	45mW	6.57mW	278mW	28.67μW	35-136mW
Frequency	1KHz-1.5MHz	67.5MHz	200MHz	125MHz	10-100MHz	200MHz	24MHz	110MHz
Supply voltage	0.4-1V	0.4-1V	1V	1.2V	0.7-1.2V	0.82-1.17V	0.375-1.1V	0.7V
Performance (TOPS/W)	1.1-9.1	3.12	0.26-10.0	1.42	/	0.21	4.65-12.08	/
Norm Performance (TOPS/W·Byte)	1.1-3.4	2.34	0.52-5.0	2.84	/	0.42	1.51-4.65	/

Table 5.1: Benchmarking table showing competitive figures-of-merit compared to similar hardware accelerators.

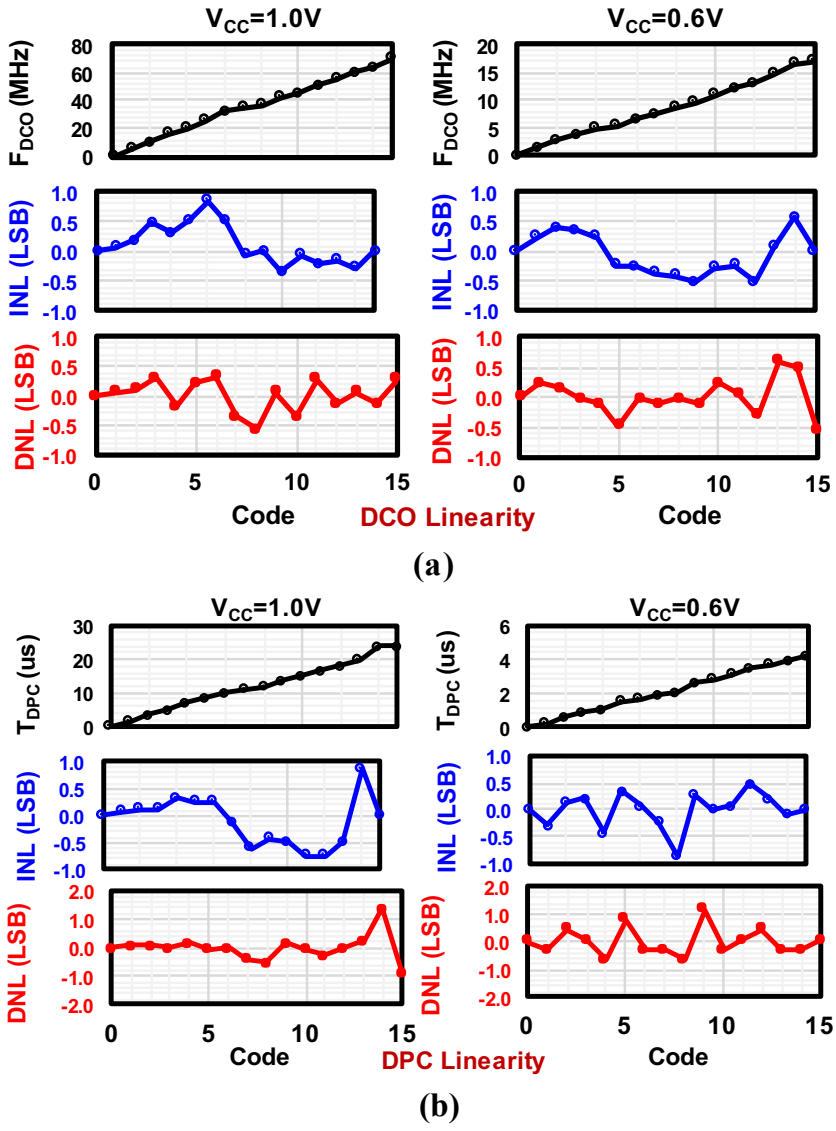


Figure 5.11: Measured linearity of (a) DCO and (b) DPC.

applications. Sample timing diagrams for two tasks, one for APF algorithm and one for the co-operative RL, are shown in Fig.5.9.

5.7 Measurements

The proposed computational platform is implemented and taped-out in 65nm GP CMOS process. It occupies a total area of $2mm^2$ and is packaged in a chip-size QFN package. The chip die photo and characteristics are shown in Fig.5.10. Since TD-MS circuits use

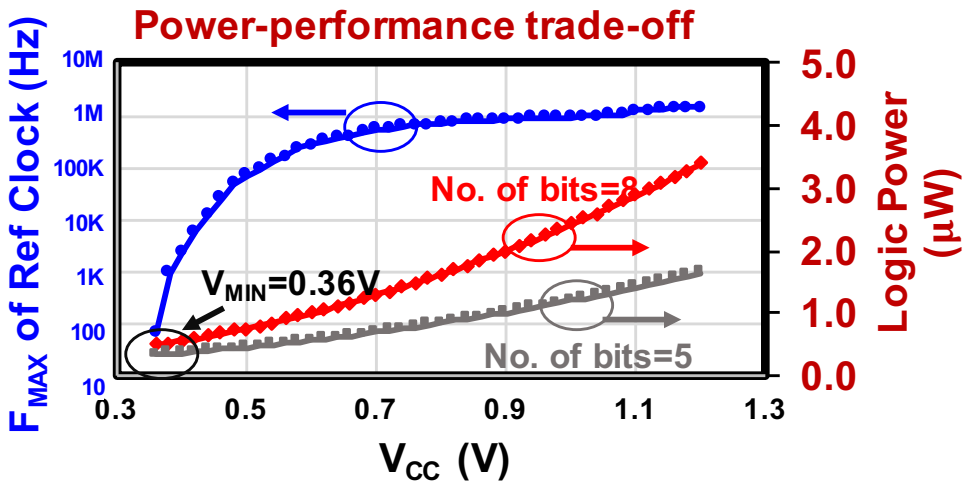


Figure 5.12: Measured power-performance trade-off.

mixed-signal DCO and DPCs, we characterize their non-linearities at two different voltages ($V_{CC} = 1.0V$ and $V_{CC} = 0.6V$). The worst case INL and DNL range from -1.0lsb to 1.1lsb, as illustrated in Fig.5.11. The measured power-performance trade-off is shown in Fig.5.12. We note a measured peak F_{MAX} of 1.5MHz and correct functionality down to a V_{MIN} of 0.36V, below which the embedded SRAM arrays cease to function. The processing throughput scales with supply voltage and thus clock frequency. We measure a logic-power dissipation 3.2 μW (1.9 μW) for 8b (5b) operations. The measured energy/op (in Fig.5.13) shows high scalability with the bit-resolution illustrating a peak of energy-efficiency of 0.22 pJ/MAC (at 3b) and 1.76 pJ/MAC (at 8b). We note that at low bit-widths, the TD-MS circuit cores show superior energy-efficiency while the digital peripherals allow almost linear energy-scaling for 5b-8b. We also measure the average arithmetic energy-efficiency as a function of the supply voltage and record a 9.1 TOPS/W (for 3b operations) and it decreases to 1.1 TOPS/W (for 8b operations) as is shown in Fig.5.14. This shows how the bit-resolution scalability allows efficient operations for multiple bit-wdths and hence swarm sizes. We plot the energy break-down of the computation unit in Fig.5.15 and show that the LPU and the NPE consumes 88% and 12% of the logic power respectively. The power-distribution across various blocks of the LPU are further shown and all the components

contribute equally in the power dissipation.

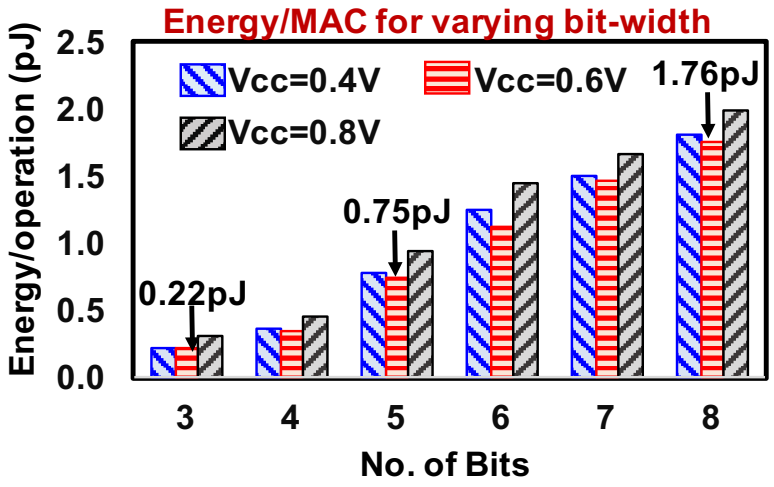


Figure 5.13: Measured energy per MAC across for different bit-widths at V_{CC} = 0.4V, 0.6V, 0.8V.

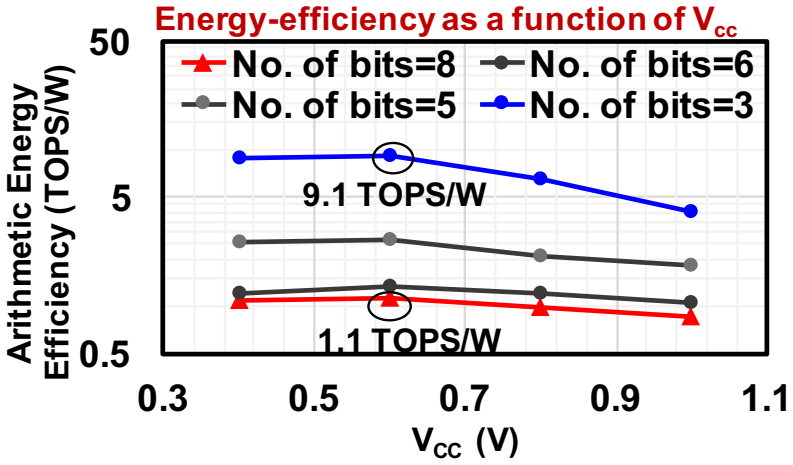


Figure 5.14: Measured arithmetic energy efficiency as a function of the operating voltage for differnet bit-widths.

The test-chip is integrated and mounted on an application platform. It is used as a controller for a robotic car as show in in Fig.5.16(a-b) and interfaces with a Raspberry-Pi, motor-controllers, sensors and LoRA radios. The convergence of co-operative RL is shown in Fig. 5.16(c). The neural-network Q-approximator has two layers and each layer has 100 neurons. Through hyper-parameter tuning, this setting results in the best performance un-

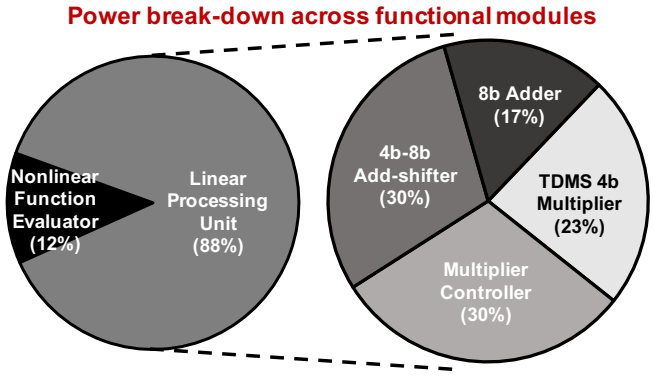


Figure 5.15: Measured power break-down among different computational blocks.

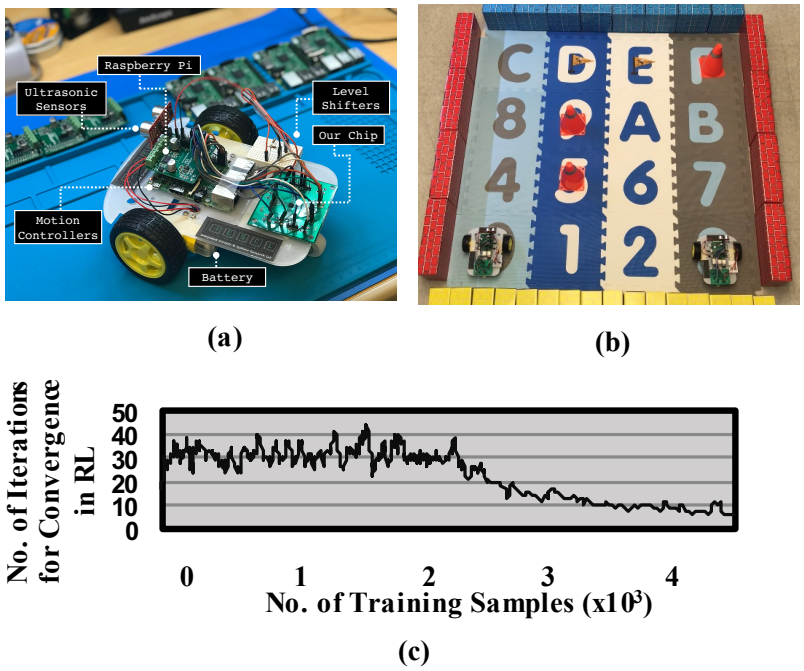


Figure 5.16: (a) Test-chip mounted on a robotic car with peripheral circuits, (b) experimental set-up and (c) the number of iterations required for convergence in co-operative RL.

der the constraints of the limited on-chip memory. Here inference is implemented in 5b TDMS and learning in 8b HDMS. In either mode, the non-linearity of the DPC and DCO (post-calibration), does not affect the accuracy of the algorithms. In particular during learning, the digital peripheral circuits for HDMS, reduces the impact of non-ideality and can successfully train the network. Further, for applications requiring higher bit-precision, the proposed HDMS can be scale to 12b-16b. A video demonstration of this can be found in:

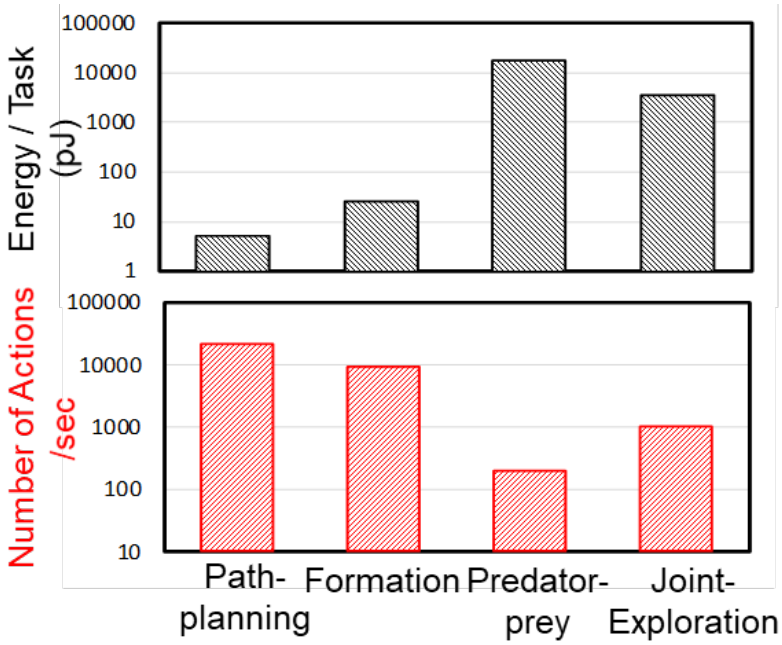


Figure 5.17: Application level benchmarking demonstrating measured energy/performance for different template algorithms.

`https://www.youtube.com/watch?v=_NqdJabFJKo`. In this video, we demonstrate a 2-robot cooperative learning task for predator-prey applications. The swarm size can also be scaled in the future for more complex demonstrations. However, the current design is limited in the number of sensor interfaces it can handle; and further modifications are required to develop real-time demonstrations of larger swarms. We implement 4 template swarm algorithms, namely: path-planning, pattern-formation, predator-prey and joint-exploration. The first two are based on physical and mathematical models and the last two are based on learning algorithms. We measure the total-energy as well as the number of actions taken per second for each of these tasks in sample environments. These are plotted in Fig.5.17 and we note a large variation in both the energy-cost and the number-of-actions-per-second for these template problems. This also illustrates the wide variety of algorithms (SLAM, vision-based path-planning and etc.) that need to be supported in future robotic controllers as the complexities of the environments and the cost-functions can dramatically change.

The test-chip has been bench-marked against similar designs and shows competitive figures-of-merit, as illustrated in Table 5.1. This is also the first demonstration of an unified and programmable platform that can accelerate a large class of algorithms for swarm robotics with efficient scalability in terms of swarm sizes and application.

5.8 Outlook

Swarm robotics is computationally challenging and the proposed test-chip is a prototype to demonstrate some key enabling features. The first challenge is scalability. The current design is limited by the on-chip memory and the number of interfaces. Future platforms can extend the design by incorporating off-chip memory to store complex "experience maps". For higher bit precision, HDMS circuits need to be evaluated, in particular, when support for more complex algorithms is required. Further,to support advanced applications, such as Simultaneous Localization And Mapping (SLAM) and vision-based navigation, the current design need to enable with higher throughput and near/in-memory computing. Finally, higher throughput can be supported through an array of LPUs and NPEs which can parallelize the algorithms.

5.9 Conclusions

This chapter presents a 65nm CMOS platform which supports both model-based and learning-based algorithms for swarm robotics. The proposed hybrid-digital-mixed-signal computational unit provides excellent scalability with swarm sizes. We measure a peak energy-efficiency of 9.1TOPS/W. The test-chip is integrated with peripheral controllers and sensors and mounted on a robotic car. Sample algorithms have been executed and bench-marked.

5.10 Discussions

5.10.1 Charge-domain vs. Time-domain

To explore energy efficiency for moderate precision computation, charge-domain computational circuit is widely applied as discussed in previous chapter. Charge-based circuit technique usually utilizes voltage to encode the data and explores per-transistor physical computation representation with continuous input/output voltage. As a result, it has advantages in power, area, delay and so on over digital counterpart for approximate computation. However, charge-domain computation has its intrinsic disadvantage. 1), Because data is encoded with voltage, the maximum achievable precision is limited by the signal to noise (SNR) ratio. As power supply scales down with Denard's scaling, it will be more difficult to sustain SNR. It has been pointed out[132] that to maintain same SNR and speed, it may even consume more energy at lower power supply. 2), Charge-domain circuit usually requires biasing to work in saturation region. In a low power system with dense analog modules, biasing current can be detrimental to overall energy efficiency. 3), As both input and output are analog, charge-domain circuit requires data conversion to interface with digital peripherals if needed. This will greatly undermines the power/area advantages brought by charge-domain computation.

For time-domain computation, it shares advantages as charge-domain computation with respect to power and area efficiency discussed in previous sections. However, as it does not require voltage to represent information, it features better ultra-low-power design potentials. 1), As data is encoded in time/frequency, voltage pulses can be pulled rail-to-rail. As a result, time-domain computation power efficiency is less affected by supply voltage scaling. 2), As voltage is rail-to-rail like digital circuit, time-domain computation does not suffer from biasing current, thus static power, as charge-domain computation. 3), time domain computation input/output are digital, so it can easily interface with digital peripherals. The HD-MS circuit technique has take advantage of it to scale TDMS computation

precision as discussed in this chapter. However, time-domain computation has its intrinsic limitations: 1), Time-domain computation is intrinsically slow. The computation latency is largely determined by the operands' magnitudes. It will not be sufficient to support real-time applications whose computation delay is critical. 2), Time-domain noise is difficult to handle. For example, there could be phase noise of pulses. The time-based noise will further prohibit time-domain computation from real-time computations. It requires extensive investigation into circuit techniques to mitigate the problem.

Overall, time-domain computation provides an alternative in approximate computation with respect to charge-domain computation. It has great potentials to implement platforms for energy/area-critical applications, such as micro-robotics, implanted medical device and so on.

5.10.2 Memory Static Power Issues in Time-domain ASIC

This test chip has demonstrated an extreme low logic power (several uW level), however, the ASIC suffers significantly from static power. The major static power consumer is the 16KB on-chip SRAM (over 70%). Although it is a severe issue for all platforms, especially nowadays with scaled technology, it is particularly problematic for time-based computation: intrinsically slow in computation, static energy dominates on average. The author has learned two implications from the issue: 1), It is crucial to have fine leakage control of SRAM banks on chip; 2), It is desirable to augment computation capability at marginal cost when static power is significant. Potential solutions are many, author will list a few. 1), Enable retention/power gating mode of SRAMs; 2), Break large SRAM into a bank of small SRAM modules and only turn on the required ones; 3), Parallel time-based computations to improve throughput or tasks latency. Proper retention/power-gating control is expected to save 40% on leakage, and a 10$\times$ parallelism is expected to shorten task-level latency thus average static energy by a similar scale. If this ASIC is going to be re-designed, the author is expecting one order of magnitude static power deduction and same order throughput

improvement.

CHAPTER 6
CONCLUSIONS

6.1 EI Expands the IoT Solution Space

Conventionally, the affordable application on IoT system is determined by each individual module's performance. For example, the processor's power/throughput efficiency, data converter speed, transmitter energy per bit and so on. However, as we discussed in previous chapters, such a discrete method can introduce a large number of design abstractions in IoT systems which in turn causes severe efficiency/information loss. This has become a bottleneck to pushing IoT-enabled applications forward.

On the contrary, EI design has largely expanded the IoT solution space in an integrated manner. Besides pushing individual module's metric to extreme, EI also tries to solve problems not in a traditional design paradigm but in a more fundamental manner. For example, in the effort to minimize ultrasonic sensor data conversion overhead or improve data conversion speed in a mirco-robot, an enhanced ADC module is mandatory. However, EI design can alternatively eliminate data conversion stage and choose to design customized time-domain circuit that is inherently compatible with ultrasonic pulses. Not only data conversion, such design methodology has motivated customized computational circuit to improve ML/AI algorithms as well. This is of particular interest when we are witnessing the slowing down of Moore's Law and the conventional digital design fails to offer sufficient improvement. The compute-in-memory module discussed in previous chapters is one example that takes advantage of circuit-algorithm co-design. This module has largely saves data access energy through bit-line charge accumulation and time-domain in-situ update for ultra-low-power inference/learning. Finally, when expanding the scope further even beyond algorithm to the application scenarios, we find a systematic solution

may bring even more benefits than individual innovations at circuits and algorithms. For example, the online computation-communication trade-off discussed in chapter III and IV is such an example. The awareness of environmental dynamics and its impact on communication cost have led to online edge-cloud trade-off. This scheme has fundamentally reshaped wireless IoT architecture. With such a system solution, we have observed 2-5 $\times$ energy/delay improvement.

To conclude, EI is about breaking conventional IoT design paradigm with customized design. In such a way, the design hierarchy is flattened and data path shortened, and most importantly, the solution space has expanded so that we can tackle IoT challenges from a fundamental design point.

6.2 EI is More Than 'Edge Computation'

Edge computation is a closely related research field to EI. However, as we have discussed previously, "intelligence" is more than "computation". The intelligence we are talking about for EI is more than ML/AI inference or learning, but also how the device as an agent reacts to the environment in a smart manner. For example, we explore how a robot intelligently controls and communicates with other robots or agents in the network. Chapter V is a control example to demonstrate edge intelligence in robotic problems. The ASIC is so designed and optimized to support machine learning, reinforcement learning and other model-based algorithms in multi-agent collaboration control. Further, chapter III-IV has visited both chip control intelligence as well as communication intelligence. We have measured significant system-level performance and adaptation enhancement by embedding intelligence into control and communication.

To conclude, for EI, we not only need the edge node to acquire data, process the data, but in the future we also would like it to take actions when required and smartly exchange data in the network. In the future, I am expecting to see more EI implementation of ML/AI in control and communication modules and looking forward to see how they interact with

each other.

6.3 Integrated Circuits to Enable EI

Solid-state chip design is an important approach to improve platform performance as we have demonstrated in Chapter IV and Chapter V. Through customized design, we are able to largely improve computation efficiency for specific algorithms, achieve real-time transmission control, reduce power and area consumption and so on.

However, custom IC solution has its disadvantages. Although with EDA tool development and technology scaling, circuit/system design have become easier and fabrication with early technology nodes have become cheaper, the relative cost and effort to fabricate/design chip compared with discrete design are still significant. One way to make the most use of chip design is to enhance versatility of the designed chip and make it compatible with varying use-cases. In chapter V, I have demonstrated a unified platform with non-linear function evaluator (NFE) and linear processing unit (LPU) to account for varying swarm robotic applications. Further, the system features bit-precision scaling to provide capability to adapt to varying swarm sizes. In chapter IV, the wireless image processing SoC has reconfigurable processing pipeline, processing element array, programmable transceiver, adaptive controller and so on. These designs in particular want to address the versatility in solid-state chip design. But versatility does not come without actual cost. For example, area and power are two major cost of versatility. For example, the digital peripherals for bit-precision scaling in HD-MS method, the controller overhead, the PE spatial array interconnection area and so on. It requires designers' expertise to improve chip use-cases with as minimal overhead as possible.

To conclude, EI chip design is costly, the versatility of the chip is desired but requires extensive circuit innovations and systematic analysis to minimize the power/area overhead.

6.4 Future EI Research Directions

Edge intelligence is an exciting and broad research field. The projects and discussions in this dissertation are far from enough to provide a comprehensive overview. During EI investigation in this dissertation, the author has been exposed to many interesting EI topics and would like to share. The author believe the successes in these research fields will make great significance in both academic and industrial world. And the author further believe these successes will eventually lead to technology development that benefit all human-kinds.

1. *Neuromorphic Microbotics*:

   Microbots often refer to small robots under 1mm dimension that is capable of handling perception, computation and actuation tasks [133]. Due to their small size and potentially very cheap cost, it is desirable to use large numbers of them (as discussed in Chapter V) to explore environments that are too small or too dangerous for people. It is expected that microbots will be very useful in disaster rescue, environmental sensing, medical care, scientific research, social study and so on. However, the computation and energy constraints are major challenges for microbots due to small form-factor. Although the individual's lack of efficiency could be partly mitigated by applying swarm robotic strategy, enhancing intelligence (by maximizing information per cost) is at the core of microbotic breakthrough.

   On the other hand, brain has been so far the most efficient computation engine on these earth. It has supreme computation capability to solve complex task with only tens of watts. Its efficiency is several orders of magnitude higher than state-of-art server engines. People believe the extreme high "information per cost" lie in its computation architecture. The computation, compared with the hardware nowadays, has features such as low speed, asynchronous, noise tolerant, spike-based, massively connected (1:10000), sparse, distributed and so on. These features have inspired

many interesting works in neuromorphic hardware to improve efficiency [134, 135].

However, the interplay between neuromorphic hardware and microbotics have not been widely explored. The extreme efficiency of brain-inspired computation will enable significant intelligence in the resource-constrained robots and the wide usage of microbots will in turn provide future direction of neuromorphic hardware in control, communication as well as computation. This dissertation, especially chapter V, has already provides some interesting insights about this research topic. In the future, it is exciting to find more interesting ideas. For example, it will be interesting to design a time-spike-based mircrobots that can process extreme low-power, real-time reinforcement learning (RL). With current edge intelligence concentrating more on perception perspective, neuromorphic microbots are expected to reveal the next phase where edge nodes make decisions and take action in a safe, secure, sustainable and autonomous manner.

2. *RF Machine Learning Systems*:

Wireless communication has become one of the major driving force of the society in fields such as IoT, multi-media, education, remote working and so on. However, the enormous end devices and ever-exploding data stream in the air has made smartness in RF system an urgent demand. In chapter III and IV, ML has been incorporated in RF system to mitigate resource constraints in highly-dynamic environment. In the future, there are other challenges to solve by exploring ML and hardware advances in RF systems. On one hand, the demands for intelligence in spectrum sensing [136] is ever-increasing. Current spectrum monitoring is on demand and it is impractical to reliably monitor and classify large bandwidth with current systems. In future, an RF system is expected to identify all signals of a given type across certain bandwidth, or so-called salience detection. At the same time, they should have the ability to exercise control over hardware receiver and extend awareness over high bandwidth (for exam-

ple higher than 10Ghz) with autonomous sensory-motor control. On the other hand, the communication security is also very important to prevent privacy risks [137]. In order to prevent the leakage and attack, it is desirable to build RF system that can apply individual discriminant physical identity. For example, how to uniquely identify a wide range of devices in a large population with feature learning, and how to learn a waveform modulation that allows for more effective discrimination with waveform synthesis.

3. *Explainable EI*: Nowadays, many ML/AI models have high performance but the decision-making mechanism remains a "blackbox", such as deep neural network and so on. In future ML/AI advances, "explanability" of the blackbox will be required for many reasons [138]. For example, to foster trust in mission-critical tasks such as medical care, to investigate in advanced DNN architecture by identifying model weakness or to bring insights to human about certain application such as Alpha Go's move to Go players are all motivations for a more interpretable model. Needless to say the regulation requirements of ML/AI data usage. There are two fields in particular that require explainable AI. First is data analytics where AI helps human analyst to look for certain items or patterns. Second is autonomy where the behavior explanations are required after each action taken by robots or autonomous cars. They all require extensive further research in explainable model and explainable interfaces.

   Explanation on the edge, or "Explainable EI", will be an extended research field that enables AI interpretability on resource-constrained IoT devices. When explanations are inserted in the data-flow pattern the current architectures for hardware-AI are broken. Support for only spatial data-flow architectures is no longer a possible solution. Data-flow processors will need to be augmented with specialized near-memory and in-memory processing elements that will concurrently provide explanability to the deep neural networks. We will develop both algorithms as well as energy-efficient hardware to enable explainable AI to augment next-generation deep convolution neu-

ral networks. The design will ensure that the results of classification and explanation are obtained by the user at the same time. Aforementioned hardware design in this dissertation is focused heavily on CNNs and DNNs for EI. However, explainable EI may require significant changes to the current architectures or even new computation paradigm. Innovations are required in memory access patterns, bit-precision for the associated explanation-logic, new logic and functional unit design to support new computational kernels and data-structures and balancing the pipeline for simultaneous decision and explanation. We will develop algorithms, models, FPGA prototypes and CMOS circuit prototypes to demonstrate key concepts.

It is expected that "explainable EI" is able to achieve both real-time ML/AI inference/learning, but also explanation. It will find broad application in autonomous vehicles, wearable medical devices and more [139].

The topics mentioned here cover various design fields, such as robotics, communication and novel ML algorithm. It is consistent with previously defined EI landscape in chapter I. At the same time, both the generic design methodology as well as some specific techniques introduced in the dissertation are helpful or provide insights into these topics.

REFERENCES

[1] *Cloud object storage for data lakes*, Wasabi, https://wasabi.com/cloud-object-storage-data-lakes/.

[2] *The IoT data explosion: How big is the IoT data market?* Priceonomics, https://priceonomics.com/the-iot-data-explosion-how-big-is-the-iot-data/.

[3] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained iot nodes: Opportunities and challenges," *IEEE Design Test*, vol. 36, no. 2, pp. 7–40, 2019.

[4] T. N. Theis and H. P. Wong, "The end of moore's law: A new beginning for information technology," *Computing in Science Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[5] K. Flamm, "Measuring moore's law: Evidence from price, cost, and quality indexes," National Bureau of Economic Research, Working Paper 24553, 2018.

[6] "The growth of lithium-ion battery power," *The Economist*, Aug. 14, 2017.

[7] F. Samie, L. Bauer, and J. Henkel, "Iot technologies for embedded computing: A survey," in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2016, pp. 1–10.

[8] K. D. Choo, L. Xu, Y. Kim, J. Seol, X. Wu, D. Sylvester, and D. Blaauw, "5.2 energy-efficient low-noise CMOS image sensor with capacitor array-assisted charge-injection SAR ADC for motion-triggered low-power IoT applications," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 96–98.

[9] C. Xu, Y. Mo, G. Ren, W. Ma, X. Wang, W. Shi, J. Hou, K. Shao, H. Wang, P. Xiao, Z. Shao, X. Xie, X. Wang, and C. Yiu, "5.1 a stacked global-shutter CMOS imager with SC-type hybrid-GS pixel and self-knee point calibration single frame HDR and on-chip binarization algorithm for smart vision applications," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 94–96.

[10] A. Amaravati, S. Xu, J. Romberg, and A. Raychowdhury, "A 130 nm 165 nJ/frame compressed-domain smashed-filter-based mixed-signal classifier for âœin-sensorâ analytics in smart cameras," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 3, pp. 296–300, Mar. 2018.

[11] S. K. Nayar, D. C. Sims, and M. Fridberg, "Towards self-powered cameras," in *2015 IEEE International Conference on Computational Photography (ICCP)*, Houston, TX, USA: IEEE, Apr. 2015, pp. 1–10, ISBN: 978-1-4799-8667-5.

[12] M. Yang, C. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "Design of an always-on deep neural network-based 1- î w voice activity detector aided with a customized software model for analog feature extraction," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1764–1777, Jun. 2019.

[13] M. Cho, S. Oh, Z. Shi, J. Lim, Y. Kim, S. Jeong, Y. Chen, D. Blaauw, H. Kim, and D. Sylvester, "17.2 a 142nw voice and acoustic activity detection chip for mm-scale sensor nodes using time-interleaved mixer-based frequency scanning," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 278–280.

[14] K. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "24.2 context-aware hierarchical information-sensing in a 6îw 90nm CMOS voice activity detector," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb. 2015, pp. 1–3.

[15] Y. Lee, K. Kim, J. Lee, K. Lee, S. Gweon, M. Kim, and H. Yoo, "17.7 a 7.0fps optical and electrical dual tomographic imaging SoC for skin-disease diagnosis system," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 288–289.

[16] N. Nesa and I. Banerjee, "IoT-based sensor data fusion for occupancy sensing using dempsterâ"shafer evidence theory for smart buildings," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1563–1570, Oct. 2017.

[17] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1514–1524, Jul. 2016.

[18] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[19] E. H. Lee and S. S. Wong, "24.2 a 2.5ghz 7.7tops/w switched-capacitor matrix multiplier with co-designed local memory in 40nm," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 418–419.

[20] J.-H. Yoon and A. Raychowdhury, "31.1 A 65nm 8.79TOPS/W 23.82mW Mixed-Signal Oscillator-Based NeuroSLAM Accelerator for Applications in Edge Robotics,"

in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, ISSN: 2376-8606, Feb. 2020, pp. 478–480.

[21] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 î j/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.

[22] A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 488–490.

[23] J. Yang, Y. Kong, Z. Wang, Y. Liu, B. Wang, S. Yin, and L. Shi, "24.4 sandwich-RAM: An energy-efficient in-memory BWN architecture with pulse-width modulation," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 394–396.

[24] M. Chang, L.-H. Lin, J. Romberg, and A. Raychowdhury, "Optimo: A 65Nm 270Mhz 143.2Mw Programmable Spatial-Array-Processor With A Hierarchical Multi-Cast On-Chip Network For Solving Distributed Optimizations," in *2019 IEEE Custom Integrated Circuits Conference (CICC)*, ISSN: 2152-3630, Apr. 2019, pp. 1–4.

[25] W. Chen, K. Li, W. Lin, K. Hsu, P. Li, C. Yang, C. Xue, E. Yang, Y. Chen, Y. Chang, T. Hsu, Y. King, C. Lin, R. Liu, C. Hsieh, K. Tang, and M. Chang, "A 65nm 1mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 494–496.

[26] M. Jerry, P. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, and S. Datta, "Ferroelectric FET analog synapse for acceleration of deep neural network training," in *2017 IEEE International Electron Devices Meeting (IEDM)*, Dec. 2017, pp. 6.2.1–6.2.4.

[27] N. Shibata, K. Kanda, T. Shimizu, J. Nakai, O. Nagao, N. Kobayashi, M. Miakashi, Y. Nagadomi, T. Nakano, T. Kawabe, T. Shibuya, M. Sako, K. Yanagidaira, T. Hashimoto, H. Date, M. Sato, T. Nakagawa, H. Takamoto, J. Musha, T. Minamoto, M. Uda, D. Nakamura, K. Sakurai, T. Yamashita, J. Zhou, R. Tachibana, T. Takagiwa, T. Sugimoto, M. Ogawa, Y. Ochi, K. Kawaguchi, M. Kojima, T. Ogawa, T. Hashiguchi, R. Fukuda, M. Masuda, K. Kawakami, T. Someya, Y. Kajitani, Y. Matsumoto, N. Morozumi, J. Sato, N. Raghunathan, Y. L. Koh, S. Chen, J. Lee, H. Nasu, H. Sugawara, K. Hosono, T. Hisada, T. Kaneko, and H. Nakamura, "13.1 a 1.33tb 4-bit/cell 3d-flash memory on a 96-word-line-layer technology," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 210–212.

[28] L. Wei, J. G. Alzate, U. Arslan, J. Brockman, N. Das, K. Fischer, T. Ghani, O. Golonzka, P. Hentges, R. Jahan, P. Jain, B. Lin, M. Meterelliyoz, J. Oâ™Donnell, C. Puls, P. Quintero, T. Sahu, M. Sekhar, A. Vangapaty, C. Wiegand, and F. Hamzaoglu, "13.3 a 7mb STT-MRAM in 22ffl FinFET technology with 4ns read sensing time at 0.9v using write-verify-write scheme and offset-cancellation sensing technique," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 214–216.

[29] S. Pellerano, S. Callender, W. Shin, Y. Wang, S. Kundu, A. Agrawal, P. Sagazio, B. Carlton, F. Sheikh, A. Amadjikpe, W. Lambert, D. S. Vemparala, M. Chakravorti, S. Suzuki, R. Flory, and C. Hull, "9.7 a scalable 71-to-76ghz 64-element phased-array transceiver module with 2ã—2 direct-conversion IC in 22nm FinFET CMOS technology," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 174–176.

[30] Z. Bai, W. Yuan, A. Azam, and J. S. Walling, "4.3 a multiphase interpolating digital power amplifier for TX beamforming in 65nm CMOS," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 78–80.

[31] S. Lee, R. Dong, T. Yoshida, S. Amakawa, S. Hara, A. Kasamatsu, J. Sato, and M. Fujishima, "9.5 an 80gb/s 300ghz-band single-chip CMOS transceiver," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 170–172.

[32] K. A. Yau, P. Komisarczuk, and P. D. Teal, "Context-awareness and intelligence in distributed cognitive radio networks: A reinforcement learning approach," in *2010 Australian Communications Theory Workshop (AusCTW)*, Feb. 2010, pp. 35–42.

[33] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 946–957, May 2012.

[34] R. Senguttuvan, S. Sen, and A. Chatterjee, "Multidimensional adaptive power management for low-power operation of wireless devices," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 9, pp. 867–871, Sep. 2008.

[35] D. Banerjee, S. Devarakond, S. Sen, and A. Chatterjee, "Real-time use-aware adaptive MIMO RF receiver systems for energy efficiency under BER constraints," in *Proceedings of the 50th Annual Design Automation Conference*, ser. DAC '13, event-place: Austin, Texas, New York, NY, USA: ACM, 2013, 56:1–56:7, ISBN: 978-1-4503-2071-9.

[36] S. Maity, K. Mojabe, and S. Sen, "Characterization of human body forward path loss and variability effects in voltage-mode HBC," *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 3, pp. 266–268, Mar. 2018.

[37] R. J. Drost, R. D. Hopkins, R. Ho, and I. E. Sutherland, "Proximity communication," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1529–1535, Sep. 2004.

[38] T. Karnik, D. Kurian, P. Aseron, R. Dorrance, E. Alpman, A. Nicoara, R. Popov, L. Azarenkov, M. Moiseev, L. Zhao, S. Ghosh, R. Misoczki, A. Gupta, M. Akhila, S. Muthukumar, S. Bhandari, Y. Satish, K. Jain, R. Flory, C. Kanthapanit, E. Quijano, B. Jackson, H. Luo, S. Kim, V. Vaidya, A. Elsherbini, R. Liu, F. Sheikh, O. Tickoo, I. Klotchkov, M. Sastry, S. Sun, M. Bhartiya, A. Srinivasan, Y. Hoskote, H. Wang, and V. De, "A cm-scale self-powered intelligent and secure IoT edge mote featuring an ultra-low-power SoC in 14nm tri-gate CMOS," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 46–48.

[39] S. Paul, V. Honkote, R. Kim, T. Majumder, P. Aseron, V. Grossnickle, R. Sankman, D. Mallik, S. Jain, S. Vangal, J. Tschanz, and V. De, "An energy harvesting wireless sensor node for IoT systems featuring a near-threshold voltage IA-32 microcontroller in 14nm tri-gate CMOS," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2.

[40] V. Honkote, D. Kurian, S. Muthukumar, D. Ghosh, S. Yada, K. Jain, B. Jackson, I. Klotchkov, M. R. Nimmagadda, S. Dattawadkar, P. Deshmukh, A. Gupta, J. Timbadiya, R. Pali, K. Narayanan, S. Soni, S. Chhabra, P. Dhama, N. Sreenivasulu, J. Kollikunnel, S. Kadavakollu, V. D. Sivaraj, P. Aseron, L. Azarenkov, N. Robinson, A. Radhakrishnan, M. Moiseev, G. Nandakumar, A. Madhukumar, R. Popov, K. P. Sahu, R. Peguvandla, A. D. R. Ruiz, M. Bhartiya, A. Srinivasan, and V. De, "2.4 a distributed autonomous and collaborative multi-robot system featuring a low-power robot SoC in 22nm CMOS for integrated battery-powered minibots," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 48–50.

[41] R. Zhou, L. Liu, S. Yin, A. Luo, X. Chen, and S. Wei, "A VLSI design of sensor node for wireless image sensor network," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 149–152.

[42] F. Samie, L. Bauer, and J. Henkel, "IoT technologies for embedded computing: A survey," in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct. 2016, pp. 1–10.

[43] A. Keshavarzi and W. v. d. Hoek, "Edge intelligenceâ"on the challenging road to a trillion smart connected IoT devices," *IEEE Design Test*, vol. 36, no. 2, pp. 41–64, Apr. 2019.

[44] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT nodes: Opportunities and challenges," *IEEE Design Test*, vol. 36, no. 2, pp. 7–40, Apr. 2019.

[45] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10, New York, NY, USA: ACM, 2010, pp. 211–224, ISBN: 978-1-4503-0344-6.

[46] T. Peffer, M. Pritoni, A. Meier, C. Aragon, and D. Perry, "How people use thermostats in homes: A review," *Building and Environment*, vol. 46, no. 12, pp. 2529–2541, Dec. 1, 2011.

[47] N. Li, G. Calis, and B. Becerik-Gerber, "Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations," *Automation in Construction*, vol. 24, pp. 89–99, Jul. 2012.

[48] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ser. BuildSys '10, New York, NY, USA: ACM, 2010, pp. 1–6, ISBN: 978-1-4503-0458-0.

[49] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: Using occupancy sensors to save energy in homes," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10, New York, NY, USA: ACM, 2010, pp. 211–224, ISBN: 978-1-4503-0344-6.

[50] N. Cao, S. Nasir, S. Sen, and A. Raychowdhury, "In-sensor analytics and energy-aware self-optimization in a wireless sensor node," in *IEEE International Microwave Symposium (IMS)*, 2017.

[51] N. Cao, B. S. Nasir, S. Sen, and A. Raychowdhury, "Self-optimizing IoT wireless video sensor node with in-situ data analytics and context-driven energy-aware real-time adaptation," *Transcation on Circuit and System, regular paper (TCAS-I), in press*,

[52] *Buildings Overview | Center for Climate and Energy Solutions*, `https://www.c2es.org/technology/overview/buildings`.

[53] Z. Yi and F. Liangzhong, "Moving object detection based on running average background and temporal difference," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Nov. 2010, pp. 270–272.

[54] J. Han and B. Bhanu, "Fusion of color and ir video for moving human detection," *Pattern Recognition*, vol. 40, no. 6, pp. 1771–1784, Jun. 2007.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun. 2005, 886–893 vol. 1.

[56] ——, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun. 2005, 886–893 vol. 1.

[57] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1491–1498.

[58] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *Journal of Biomedical Informatics*, vol. 35, no. 5, pp. 352–359, Oct. 2002.

[59] "Collaborative intelligence in optical/ir camera based wireless sensor nodes for hvac control," *IEEE Sensors 2017 Conference*, no. 5, Nov. 2017.

[60] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, Jan. 1997.

[61] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, Jan. 2013.

[62] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[63] F. Ren, T. He, S. K. Das, and C. Lin, "Traffic-Aware Dynamic Routing to Alleviate Congestion in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1585–1599, Sep. 2011.

[64] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03, New York, NY, USA: ACM, 2003, pp. 266–279, ISBN: 978-1-58113-707-1.

[65] T. Pal, S. Bandyopadhyay, and S. Dasbit, "Energy-Saving Image Transmission over WMSN Using Block Size Reduction Technique," in *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, Dec. 2015, pp. 41–46.

[66] H. C. Tijms, *A First Course in Stochastic Models*. Wiley, May 2003, Google-Books-ID: eBeNngEACAAJ, ISBN: 978-0-471-49881-0.

[67] Z. Yi and F. Liangzhong, "Moving object detection based on running average background and temporal difference," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Nov. 2010, pp. 270–272.

[68] S. Soyguder, M. Karakose, and H. Alli, "Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4566–4573, Apr. 1, 2009.

[69] S. Soyguder and H. Alli, "An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with fuzzy modeling approach," *Energy and Buildings*, vol. 41, no. 8, pp. 814–822, Aug. 1, 2009.

[70] S. Servet and A. Hasan, "Predicting of fan speed for energy saving in HVAC system based on adaptive network based fuzzy inference system," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8631–8638, May 1, 2009.

[71] S. Soyguder, "Intelligent system based on wavelet decomposition and neural network for predicting of fan speed for energy saving in HVAC system," *Energy and Buildings*, vol. 43, no. 4, pp. 814–822, Apr. 1, 2011.

[72] V. L. Erickson, M. Carreira-Perpiñán, and A. E. Cerpa, "Observe: Occupancy-based system for efficient reduction of hvac energy," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Apr. 2011, pp. 258–269.

[73] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy based hvac actuation using existing wifi infrastructure within commercial buildings," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '13, New York, NY, USA: ACM, 2013, 17:1–17:14, ISBN: 978-1-4503-2027-6.

[74] S. Ehsan and B. Hamdaoui, "A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

[75] Z. Takhirov, J. Wang, V. Saligrama, and A. Joshi, "Energy-Efficient Adaptive Classifier Design for Mobile Systems," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ser. ISLPED '16, New York, NY, USA: ACM, 2016, pp. 52–57, ISBN: 978-1-4503-4185-1.

[76] A. Anvesha, S. Xu, N. Cao, J. Romberg, and A. Raychowdhury, "A Light-powered, "Always-On", Smart Camera with Compressed Domain Gesture Detection," in

Proceedings of the 2016 International Symposium on Low Power Electronics and Design, ser. ISLPED '16, New York, NY, USA: ACM, 2016, pp. 118–123, ISBN: 978-1-4503-4185-1.

[77] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[78] A. D. Wood, J. A. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru, "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Network*, vol. 22, no. 4, pp. 26–33, Jul. 2008.

[79] S. Sen, R. Senguttuvan, and A. Chatterjee, "Environment-Adaptive Concurrent Companding and Bias Control for Efficient Power-Amplifier Operation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 607–618, Mar. 2011.

[80] M. K. Chowdary, S. S. Babu, S. S. Babu, and H. Khan, "FPGA implementation of moving object detection in frames by using background subtraction algorithm," in *2013 International Conference on Communication and Signal Processing*, Apr. 2013, pp. 1032–1036.

[81] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[82] K. Kinoshita, M. Enokidani, M. Izumida, and K. Murakami, "Tracking of a Moving Object Using One-Dimensional Optical Flow with a Rotating Observer," in *Robotics and Vision 2006 9th International Conference on Control, Automation*, Dec. 2006, pp. 1–6.

[83] S. McCann and D. G. Lowe, "Local Naive Bayes Nearest Neighbor for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 3650–3656.

[84] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground Object Detection from Videos Containing Complex Background," in *Proceedings of the Eleventh ACM International Conference on Multimedia*, ser. MULTIMEDIA '03, New York, NY, USA: ACM, 2003, pp. 2–10, ISBN: 978-1-58113-722-4.

[85] D. J. Moore, I. A. Essa, and M. H. Hayes, "Exploiting human actions and object context for recognition tasks," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, 1999, 80–86 vol.1.

[86] B. Wu and R. Nevatia, "Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.

[87] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sensing of Environment*, vol. 61, no. 3, pp. 399–409, Sep. 1997.

[88] D. Banerjee, S. Sen, and A. Chatterjee, "Self learning analog/mixed-signal/RF systems: Dynamic adaptation to workload and environmental uncertainties," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2015, pp. 59–64.

[89] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A Power-Scalable Channel-Adaptive Wireless Receiver Based on Built-In Orthogonally Tunable LNA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 946–957, May 2012.

[90] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-Variation Tolerant Channel-Adaptive Virtually Zero-Margin Low-Power Wireless Receiver Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1764–1777, Dec. 2014.

[91] S. Sen, V. Natarajan, R. Senguttuvan, and A. Chatterjee, "Pro-VIZOR: Process tunable virtually zero margin low power adaptive RF for wireless systems," in *2008 45th ACM/IEEE Design Automation Conference*, Jun. 2008, pp. 492–497.

[92] S. Sen, "Invited - Context-aware Energy-efficient Communication for IoT Sensor Nodes," in *Proceedings of the 53rd Annual Design Automation Conference*, ser. DAC '16, New York, NY, USA: ACM, 2016, 67:1–67:6, ISBN: 978-1-4503-4236-0.

[93] D. Banerjee, B. Muldrey, S. Sen, X. Wang, and A. Chatterjee, "Self-learning MIMO-RF receiver systems: Process resilient real-time adaptation to channel conditions for low power operation," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 710–717.

[94] D. Banerjee, B. Muldrey, X. Wang, S. Sen, and A. Chatterjee, "Self-Learning RF Receiver Systems: Process Aware Real-Time Adaptation to Channel Conditions for Low Power Operation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, no. 99, pp. 1–13, 2016.

[95] S. Sen, R. Senguttuvan, and A. Chatterjee, "Concurrent PAR and power amplifier adaptation for power efficient operation of WiMAX OFDM transmitters," in *2008 IEEE Radio and Wireless Symposium*, Jan. 2008, pp. 21–24.

[96] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, Feb. 1992.

[97] M. Imran, K. Shahzad, N. Ahmad, M. O'Nils, N. Lawal, and B. Oelmann, "Energy-efficient SRAM FPGA-based wireless vision sensor node: SENTIOF-CAM," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2132–2143, Dec. 2014.

[98] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1438–1452, Oct. 2011.

[99] M. Imran, K. Khursheed, N. Lawal, M. O'Nils, and N. Ahmad, "Implementation of wireless vision sensor node for characterization of particles in fluids," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 11, pp. 1634–1643, Nov. 2012.

[100] L. Zhang, D. Fu, J. Liu, E. C. H. Ngai, and W. Zhu, "On energy-efficient offloading in mobile cloud for real-time video applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 170–181, Jan. 2017.

[101] S. Sen, R. Senguttuvan, and A. Chatterjee, "Environment-adaptive concurrent companding and bias control for efficient power-amplifier operation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 607–618, Mar. 2011.

[102] Z. Shi, J. Tu, Q. Zhang, L. Liu, and J. Wei, "A Survey of Swarm Robotics System," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and Z. Ji, Eds., ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 564–572, ISBN: 978-3-642-30976-2.

[103] H. M. La, R. Lim, and W. Sheng, "Multirobot Cooperative Learning for Predator Avoidance," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, Jan. 2015.

[104] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

[105] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[106] J. Pugh and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in *2007 IEEE Swarm Intelligence Symposium*, 2007, pp. 332–339.

[107] D. P. Stormont, "Autonomous rescue robot swarms for first responders," in *CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, 2005.*, Mar. 2005, pp. 151–157.

[108] S. Taranovich, *Efficient powering of a robot swarm | EDN*, `https://www.edn.com/design/power-management/4442493/Efficient-powering-of-a-robot-swarm`, Aug. 2016.

[109] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 124–126.

[110] C. W. Warren, "Multiple robot path coordination using artificial potential fields," in *, IEEE International Conference on Robotics and Automation Proceedings*, May 1990, 500–505 vol.1.

[111] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2, Jul. 2003, 735–740 vol.2.

[112] J. Barraquand, B. Langlois, and J. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, Mar. 1992.

[113] P. Vadakkepat, Kay Chen Tan, and Wang Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, 2000, 256–263 vol.1.

[114] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing," in *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, 2006, pp. 622–627.

[115] P. N. Whatmough et al., "A 28nm SoC with a 1.2GHz 568nJ/Prediction sparse deep-neural-network engine with > 0.1 timing error rate tolerance for IoT applications," *ISSCC*, 2017.

[116] P. N. Whatmough et al., "Dnn engine: A 28-nm timing-error tolerant sparse deep neural network processor for iot applications," *IEEE Journal of Solid State Circuits*, vol. 53, no. 9, September 2018.

[117] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, Jun. 2016, pp. 367–379.

[118] L. Bu¿oniu, R. B. \. s$ka, and B. D. Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[119] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine, "Collective robot reinforcement learning with distributed asynchronous guided policy search," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 79–86.

[120] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–8.

[121] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning," *Machine Learning*, vol. 23, no. 2, pp. 279–303, May 1996.

[122] M. A. Anwar and A. Raychowdhury, "NavREn-Rl: Learning to fly in real environment via end-to-end deep reinforcement learning using monocular images," in *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nov. 2018, pp. 1–6.

[123] S. Peng, "A Generalized dynamic programming principle and hamilton-jacobi-bellman equation," *Stochastics and Stochastic Reports*, vol. 38, no. 2, pp. 119–134, Feb. 1992.

[124] A. Amaravati, S. B. Nasir, J. Ting, I. Yoon, and A. Raychowdhury, "A 55-nm, 1.0–0.4v, 1.25-pJ/MAC Time-Domain Mixed-Signal Neuromorphic Accelerator With Stochastic Synapses for Reinforcement Learning in Autonomous Mobile Robots," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 75–87, Jan. 2019.

[125] N. Cao, M. Chang, and A. Raychowdhury, "14.1 A 65nm 1.1-to-9.1tops/W Hybrid-Digital-Mixed-Signal Computing Platform for Accelerating Model-Based and Model-Free Swarm Robotics," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 222–224.

[126] A. Sayal, S. Fathima, S. S. T. Nibhanupudi, and J. P. Kulkarni, "14.4 All-Digital Time-Domain CNN Engine Using Bidirectional Memory Delay Lines for Energy-Efficient Edge Computing," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 228–230.

[127] D. Miyashita, R. Yamaki, K. Hashiyoshi, H. Kobayashi, S. Kousai, Y. Oowaki, and Y. Unekawa, "An LDPC Decoder With Time-Domain Analog and Digital Mixed-Signal Processing," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 73–83, Jan. 2014.

[128] Z. Chen and J. Gu, "19.7 A Scalable Pipelined Time-Domain DTW Engine for Time-Series Classification Using Multibit Time Flip-Flops With 140giga-Cell-Updates/s Throughput," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 324–326.

[129] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 Envision: A 0.26-to-10tops/W subword-parallel dynamic-voltage-accuracy-frequency-scalable Convolutional Neural Network processor in 28nm FDSOI," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2017, pp. 246–247.

[130] J. Sim, J. Park, M. Kim, D. Bae, Y. Choi, and L. Kim, "14.6 A 1.42tops/W deep convolutional neural network recognition processor for intelligent IoE systems," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 264–265.

[131] S. Choi, J. Lee, K. Lee, and H. Yoo, "A 9.02mw CNN-stereo-based real-time 3d hand-gesture recognition processor for smart mobile devices," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 220–222.

[132] P. R. Kinget, "Scaling analog circuits into deep nanoscale cmos: Obstacles and ways to overcome them," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*, 2015, pp. 1–8.

[133] H. Ceylan, J. Giltinan, K. Kozielski, and M. Sitti, "Mobile microrobots for bioengineering applications," *Lab on a Chip*, vol. 17, no. 10, pp. 1705–1724, 2017.

[134] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," *arXiv:1705.06963 [cs]*, May 19, 2017. arXiv: 1705.06963.

[135] J. Hsu, "IBM's new brain [news]," *IEEE Spectrum*, vol. 51, no. 10, pp. 17–19, Oct. 2014.

[136] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, Jan. 2019.

[137] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based RF fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, Jan. 2020.

[138] W. Samek and K.-R. Müller, "Towards explainable artificial intelligence," *arXiv:1909.12072 [cs]*, vol. 11700, pp. 5–22, 2019. arXiv: 1909.12072.

[139] M. Hofmarcher, T. Unterthiner, J. Arjona-Medina, G. Klambauer, S. Hochreiter, and B. Nessler, "Visual scene understanding for autonomous driving using semantic segmentation," in, Sep. 10, 2019, pp. 285–296, ISBN: 978-3-030-28953-9.

VITA

Ningyuan Cao received his B.S. degree from Shanghai Jiaotong University and his M.S. degree from Columbia University, both in electrical engineering. He joined Integrated Circuit and System Research Lab (ICSRL) in Georgia Institute of Technology since 2015 and is currently pursuing his Ph.D. degree. His research interests include low power wireless sensor and edge intelligence platform design.