Selective CS: An Energy-Efficient Sensing Architecture for Wireless Implantable Neural Decoding

Chen Song, *Student Member, IEEE,* Aosen Wang, *Student Member, IEEE,* Feng Lin, *Member, IEEE,* Jian Xiao, *Member, IEEE,* Xinwei Yao, *Member, IEEE,* and Wenyao Xu, *Member, IEEE*

***Abstract*—The spike classification is a critical step in the implantable neural decoding. The energy efficiency issue in the sensor node is a big challenge for the entire system. Compressive sensing (CS) theory provides a potential way to tackle this problem by reducing the data volume on the communication channel. However, the constant transmission of the compressed data is still energy-hungry. On the other hand, the feasibility of direct analysis in compression domain is mathematically demonstrated. This advance empowers the in-sensor light-weight signal analysis on the compressed data. In this paper, we propose a novel selective CS architecture for energy-efficient wireless implantable neural decoding based on compression analysis and deep learning. Specifically, we develop a two-stage classification procedure, including a light-weight coarse-grained screening module in the sensor and an accurate fine-grained analysis module in the server. To achieve better energy efficiency, the screening module is designed by the Softmax regression, which can complete the low-effort classification task at the sensor end and screen the high-effort task to transmit their compressed measurements to the remote server. The fine-grained analysis located in server end is constructed by the customized deep residual neural network. It can not only promote the spike classification accuracy, but also benefit the model quality of in-sensor Softmax model. The extensive experimental results indicate that our proposed selective CS architecture can gain more than 60% energy savings than the conventional CS architecture, yet even improve the accuracy of state-of-the-art CS architectures.**

***Index Terms*—Compressed Sensing; Energy-Efficient Architecture; Deep Learning.**

I. Introduction

The spike detection in neural decoding is one of the most significant procedures to investigate and understand the neural activity mechanisms [1], [2]. Meanwhile, wireless and implantable sensor network technologies are widely explored to facilitate the patient-centric tele-monitoring and improve the quality of life [3], [4]. The typical wireless implantable systems for spike classification comprise an Electroencephalography (EEG) signal acquisition sensor and a remote processing center. However, the huge amount of data exchanging in the wireless communication makes the sensor design face with severe energy efficiency issues. Compressive sensing (CS) is a promising way to reduce the front-end data volume in that its sampling is proportional to the signal information, breaking the conventional Shannon-Nyquist rule on the data sampling [5], [6]. Recent works proved that the CS-based sampling rate can reduce a significant portion of energy consumption in data processing and wireless communication [7], [8]. However, constant compressed data transmission is still energy-hungry and how to further increase the energy efficiency in the wireless implantable applications remains a big challenge.

There are some prior works on spike classification and CS architecture, which targets on biosignals processing. Zhang *et al.* proposed a compact microsystem to apply CS into the implantable neural recording [9]. Suo *et al.* optimized the neural recording system by a two-stage sensing and a sparsifying dictionary [10]. Fallahzadeh *et al.* designed a novel adaptive compressed sensing architecture for activity recognition [11]. However, all these works just split the signal recording and classification into two independent procedures, and only concentrate on the single-facet optimization. This also constrains the integral combination of the signal recording and analysis procedure.

Meanwhile, direct analysis in the compression domain provides a theoretical cue to support the light-weight analysis computing occurring in the energy-constrained sensor-end. Direct analysis of compressed data can significantly reduce the computational cost [12], [13]. One successful example is that special visual features can be extracted from MPEG video files for understanding physical properties of video contents [14]. The random projection based compression analysis is also explored, where the text and image are successfully classified [15]. Since the birth of Compressed Sensing, the mathematicians have been exploring the possibility of analyzing compressively sensed data without reconstruction [16]. Shoaib *et al.* have even proved that direct analysis can obtain better results on unreconstructable compressed data which doesn't satisfy the RIP constraint [17]. In recent years, some researchers start to implement the compressed analysis algorithms into the front-end hardware in wearable medical applications [18].

To make up the occasional error from the light-weight signal analysis in the compression domain, an inference unit with super discrimination ability is necessary to be employed at

C. Song, A. Wang, and W. Xu are with the Department of Computer Science and Engineering, the State University of New York (SUNY) at Buffalo, NY, USA, 14260. email: ({aosenwan,csong5,wenyaoxu}@buffalo.edu).

F. Lin is with the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO, USA, (email: feng.2.lin@ucdenver.edu).

J. Xiao is with the School of Electronics and Control Engineering, Chang'an University, Xi'an, China (email: xiaojian@chd.edu.cn).

X. Yao is with the Department of Computer Science and Engneering, Zhejiang University of Technology, Hangzhou, China. (email: xwyao@zjut.edu.cn).

the server end. Deep learning [19] is widely emerging in recent years due to its remarkable ability to sense the potential rule behind the big data. Thanks to the parallel computing on GPU and advanced training skills, the current deep learning framework can support an extremely deep network with surprising learning ability. A lot of deep learning models are designed and improved in computer vision domain, such as image classification, object tracking and image captioning. The performance of deep learning is also examined on biomedical signals [20].

In this paper, we propose a novel selective CS architecture to combine the signal acquisition and partial analysis into the front-end sensing architecture for energy-efficient neural decoding system. Our entire design is driven by compression domain analysis and deep learning algorithm. Instead of always transmitting the compressed data, our new architecture integrates the signal recording and analysis, and only transmits the bit-wise analysis result when certain criteria are met. Specifically, we design a light-weight coarse-grained screening module to evaluate the classification effort in the sensor end. Given the sensor data, the architecture can conduct either the coarse-grained screening module or the fine-grained analysis modules, depending on a confidence score. For the former one, we propose to use Softmax regression model by considering the complexity and hardware-friendly requirement. For the latter one, we deploy the deep learning algorithm to execute the high-effort analysis at the server end. To keep the state-of-the-art accuracy, we particularly customize the deep residual neural network in the fine-grained analysis module. By seamlessly cooperating the two proposed modules, the selective CS architecture can provide more than 60% energy savings for the wireless implantable neural decoding. The spike classification accuracy is even improved. It's worth to point out that most physiological signals are proved to be applicable with CS (e.g., Electroencephalography [21], Electrocardiography [7] and Electromyography [22]). Therefore, our architecture can be expanded to other applications in bio and health-related sensing.

The contribution of our work can be summarized in three folds:

- We propose a new selective CS architecture to combine the signal acquisition and partial analysis into the sensor end to achieve better energy-efficiency for the neural decoding system.
- We implement our proposed architecture based on deep learning. The proposed customized technique can benefit both the coarse-grained screening and fine-grained analysis modules.
- We evaluate our model through extensive experiments with regard to the energy consumption and the spike classification accuracy. We further discuss the superiority of our design by exploiting the quality of critical design freedoms.

The remainder of this paper is organized as follows: Section II introduces the preliminaries and backgrounds. The details of our proposed selective CS architecture is discussed in Section III, and Section IV presents our experiments and evaluations to examine the selective architecture. The paper is concluded in Section V.

II. Backgrounds and Preliminaries

A. Compressed Sensing Theory

The compressive sensing theory is a new emerging analog-to-information sampling scheme. We assume that x is an N-dimension vector and is sampled using M-measurement vector y:

$$y = \Phi x, \tag{1}$$

where $\Phi \in R^{M \times N}$ is the sensing array and M is defined as the sampling rate. The elements in Φ are random variables. Because of $M << N$, the formulation in Eq. (1) is undetermined. However, under certain sparsity-inducing basis $\Psi \in R^{N \times N}$, the signal x can be represented by a set of sparse coefficients $u \in R^N$:

$$x = \Psi u. \tag{2}$$

Therefore, based on Eq. (1) and (2), the sparse vector, u, can be represented as follows:

$$y = \Phi \Psi u = \Theta_{M \times N} u, \tag{3}$$

where $\Theta_{M \times N} = \Phi \Psi$ is an $M \times N$ matrix, called the measuring matrix. In practical applications, original signals need to be quantized for transmitting. Then the compressed signal, y, is processed by a quantization model formulated as follows:

$$\widehat{y} = Q_b(y), \tag{4}$$

where $Q_b(.)$ is the quantization function [23], [21] , and $\widehat{y}$ is the quantized representation of y with b bits.

Due to the prior knowledge that the unknown vector u is sparse, u can be estimated by ℓ_1 minimization to approximate the optimal ℓ_0 minimization formulation as follows:

$$\widehat{u} = min \ \|u\|_1 \quad s.t. \quad \|\widehat{y} - \Theta u\| < \epsilon, \tag{5}$$

where ϵ is the reconstruction error margin. The ℓ_1 minimization is convex and can be solved within the polynomial time. Therefore, the reconstructed signal, $\widehat{x}$, is retrieved by:

$$\widehat{x} = \Psi \widehat{u}. \tag{6}$$

B. Compression Domain Analysis in Sensor

Many researchers paid effort on exploring the opportunity of low-power sensor node design based on CS paradigm. In the beginning, only data compression and quantization were employed in the front-end (sensor node), while the signal reconstruction and machine learning techniques were executed in the back-end (aggregator with powerful computing capability) [24]. The main task was to develop the new reconstruction algorithm to extend the lower bound of sample number while keeping data analysis accuracy [9]. Some others were also working on the energy-efficient wireless communication protocols [25].

Subsequently, in-sensor signal processing was investigated. Mathematicians demonstrated the possibility of directly analyzing the compression data [26]. Random projection can preserve signal intrinsic information from high-dimensional raw

data to low-dimension compressed representation [15]. This empowered us to avoid the computation-expensive reconstruction step and integrate the machine learning procedure into the front-end. In this way, the system can drastically reduce the communication burden by only sending the analysis result instead of the entire data. Thus, the low-power compressed-domain computing engine with the entire analysis ability [27] became a hot topic. All kinds of the platform are explored for the feasibility. The general-purpose computing devices are not suitable for computing engine design due to limits of power or resources. For example, low-power micro-processor, represented by ARM M-series, is not ready for very tight power applications [28], and the micro-controller as TI msp430 are equipped with too few resources to support machine learning implementations [29]. Therefore, application-specific integrated circuits (ASIC) stood out by its low-power consumption, which can make the power overhead of computing engine in front-end decrease to about 100uW [30]. However, the popular idea until this time was still to implement the analysis logic as a whole.

C. Deep Learning

Deep learning [19] becomes a prevalent machine learning approach in recent years. It makes the great breakthrough compared with the traditional algorithm in many applications, such as computer vision [31], speech recognition [32] and text analysis [33]. Deep learning organizes "deep" neural networks to process the input data. Its layer-by-layer scheme makes deep learning digest the data features from the low level to the high level. Therefore, deep learning can learn effective data representation from big data to enable better application performance.

Deep learning originates from the traditional neural network [34]. Due to the constrains from data size, computing ability and advanced training skills, the neural network was not becoming "deep". With the development of Imagenet large scale visual recognition challenge (ILSVRC) [35], deep learning steps into the people's sight and shows its superiority on the discrimination ability. In 2012, Krizhevsky *et al.* designed the first deep learning framework [36] to get the champion of the challenge with a large gap to the second place. This work opened the door of deep learning. Besides the big labeled high-quality data, it also brought the new training skills to suppress the overfitting problem effectively, including ReLU activation function, dropout and local response normalization. It also employed GPU as its computing platform to achieve training time breakthrough. Subsequently, VGG [37] and GoogLenet [38] were proposed to further optimize the deep neural network architectures. Their common goal is to implement "deeper" network to improve the network performance. Some advanced training stills were also developed, such as batch normalization [39] and auxiliary classifier. Recently, the deep residual network was proposed to even advance the network discrimination capability by residual design [40]. This idea solved the gradient varnishing problem in the deep learning and extend the network layer to more than 1000. All these deep neural network improvements will provide us a good reference to design our selective CS architecture.

III. Selective Compressed Sensing Architecture

In this section, we present our proposed selective CS architecture for neural decoding. Specifically, we design a coarse-grained screening module to judge the effort of signal identification in the front-end. Correspondingly, we adopt a robust deep learning algorithm in the fine-grained analysis module to process the high-effort task in the server end.

A. Architecture Overview

The selective CS architecture for wireless implantable neural decoding is illustrated in Figure 1. The entire architecture includes a low-power CS front-end and a remote server. Our ultimate goal is to optimize the energy efficiency in the front-end (the sensor node) while minimizing the accuracy compromise of signal analysis.

The front-end design comprises *three* key components, a random encoding module, a quantization module and a coarse-grained screening module. As shown in Figure 1, analog N-dimension raw sensor signal x is compressed into M-dimension measurements y in the random encoding module. The random encoding module consists of M branches with each completing a randomized combination for one measurement. Every branch includes a multiplier, a column vector in sensing array Φ and an integrator to accumulate the intermediate results. In the quantization module, there are b comparators and a digital encoder. Each comparator outputs a binary decision of comparing the input analog signal and a reference voltage level. The digital encoder organizes the final quantization result $\widehat{y}$ based on these comparison decisions. The newly proposed coarse-grained screening module analyzes the compressed measurements $\widehat{y}$. It outputs the category prediction and a confidence score. If the score is larger than the predefined threshold, the wireless transmitter only needs to send the final prediction to the remote server instead of the entire compressed data. Only when the score is below the threshold does it stream the compressed measurements $\widehat{y}$ to the server for fine-grained analysis by deep learning algorithm. In the server end, the decoder first copes with the data acquired by the wireless receiver. If the data indicates the softmax classification is reliable in the sensor-end, it will bypass the final prediction to the neural recording module. Otherwise, it will transmit the intermediate compressed measurements to the deep learning module for fine-grained analysis. Afterward, the final prediction is sent to the neural recording.

Note that the coarse-grained screening module and fine-grained analysis module are two most significant components to improve the energy efficiency of the CS front-end, yet reserving the classification accuracy. In the following section, we continue to discuss the design of these two modules in detail.

B. Coarse-Grained Screening Module

1) Softmax Prediction: It is important to have a reliable clue to determine the effort of signal category prediction in the front-end. Some low-effort tasks can be completed in the sensor node, avoiding the energy overhead of transmitting the compressed data. To this end, we consider the softmax

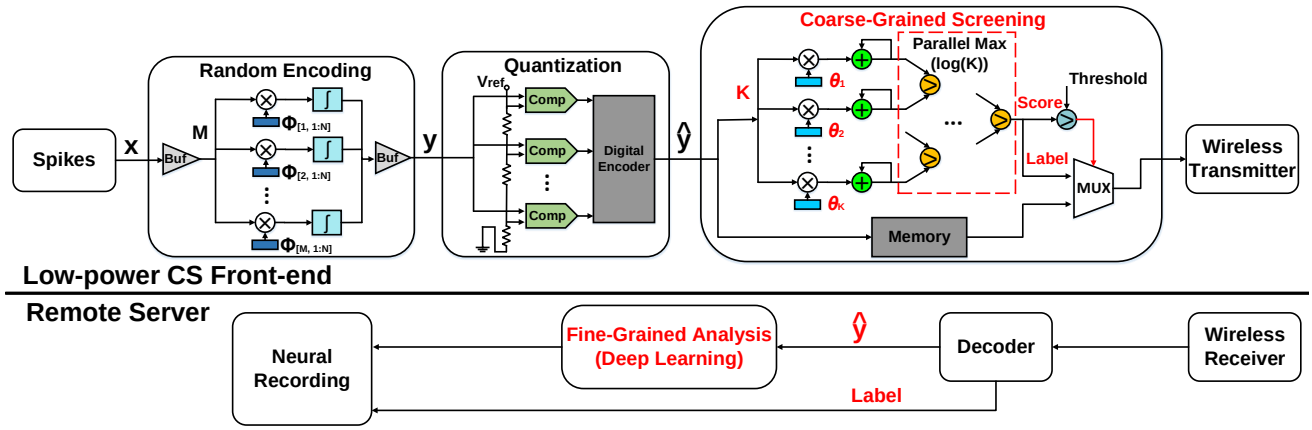

Fig. 1. The block diagram of the selective CS architecture for energy-efficient wireless implantable neural decoding, which comprises a coarse-grained screening module in the front-end and a fine-grained analysis module in the back-end. The former one conducts the energy-efficient softmax prediction upon the compressed data. If the generated confidence score is below the threshold, the latter module performs the fine-grained analysis based on deep-learning.

regression, a probabilistic model, to construct the coarse-grained screening module.

The softmax regression [41] is the extensive form of logistic regression to deal with multi-class classification problem. As the logistic regression, the hypothesis function h_θ for the softmax regression outputs a probability vector:

$$h_\theta(z) = \begin{bmatrix} P(l=1|z,\theta) \\ P(l=2|z,\theta) \\ \vdots \\ P(l=K|z,\theta) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\theta_k z}} \begin{bmatrix} e^{\theta_1 z} \\ e^{\theta_2 z} \\ \vdots \\ e^{\theta_K z} \end{bmatrix}, \quad (7)$$

where θ is the weight matrix, denoted as $\theta = [\theta_1, \theta_2, \ldots, \theta_K]$. Each weight component θ_j is a weight vector as in the logistic regression. It is reasonable to identify the input signal as the category with the largest conditional probability. Let z_i be the input feature vector in the training set, and l_i be its corresponding multi-class label, which ranges from 1 to K. The cost function to evaluate the hypothesis function h_θ is:

$$J(\theta) = -\sum_{i}\sum_{j=1}^{K}(l_i == j) log(\frac{e^{\theta_j z}}{\sum_{k=1}^{K} e^{\theta_k z}}), \quad (8)$$

where $l_i == j$ is to judge the equality. If l_i is not equal to the label j, the judge is false, denoted as "0". Otherwise, the judge is true, as "1". Similarly, we can use the gradient descent algorithm to minimize this cost function to search for the optimal weight matrix θ. The derivative of the cost function with respect to the specific weight vector θ_k is as follows:

$$\nabla_{\theta_j} J(\theta) = -\sum_{i}[z_i((l_i == j) - P(l_i = j|z_i, \theta))]. \quad (9)$$

This training phase is computation-intensive, and will be accomplished offline. Therefore, we implement the prediction phase into the front-end to analyze the compressed measurements with the pre-trained parameters. The formal formula of the prediction phase is as the following:

$$P(l = j|z, \theta) = \frac{e^{\theta_j z}}{\sum_{k=1}^{K} e^{\theta_k z}}, \quad (10)$$

when z is given, the l with the maximal probability is the prediction category. We also take this maximal probability as the confidence score S. It is reasonable that larger score has more probability of correct classification.

The prediction phase of Softmax has low hardware complexity yet robust performance, as the coarse-grained screening module shown in Figure 1. When compressed measurements come, the screening module has two parallel procedures, one is to store the input vector and the other is to start the prediction of Softmax method. In the prediction procedure, We first calculate the score $e^{\theta_j z}$ by parallel chains, where each chain is equipped with a multiplier, an accumulator and an exponential calculator. The super computation of the exponential calculator is implemented by the CORDIC algorithm [42]. Then the parallel max structures compute the maximal probability and record the related category result.

2) Confidence Score for Screening: After we obtain the category result, we also would like to know the effort to this classification task. We propose to apply confidence score to quantitatively measure this effort. The formula of the confidence score is defined as the following:

$$S = P(l = j_{max}|z, \theta) - P(l = j_{sub}|z, \theta), \quad (11)$$

where the j_{max} corresponds to the category whose probability is the maximal one and the j_{sub} is the category with probability as the second maximal. According to this equation, we can find that the confidence score S indicates the distance between the most probable one with the other options. The larger the confidence score is, the less the effort is needed to classify the input.

In practice, we set a pre-defined threshold th_s to determine whether the current effort is large enough to be processed by the fine-grained analysis module. If the confidence score is larger than the threshold, the selector, MUX module, trusts the softmax classification and chooses the prediction result for the wireless transmission. Otherwise, the MUX outputs the compressed measurements buffed in the memory. This screening can greatly reduce the energy consumption of sensor node by processing the low-effort input before the wireless

transmission.

C. Fine-Grained Analysis Module

When the coarse-grained screening module bypasses the compressed measurements to the remote server, a fine-grained classifier is highly required for the accurate classification without energy concerns. To this end, we propose to design our fine-grained classifier based on deep residual network (ResNet) [40], which is the state-of-the-art deep learning algorithm in the computer vision domain. The ResNet is the most popular algorithm in deep learning domain with the superior performance to the other machine learning approach. It even empowers the discrimination capability exceeds the humans on 1000-category image classification on the large dataset.

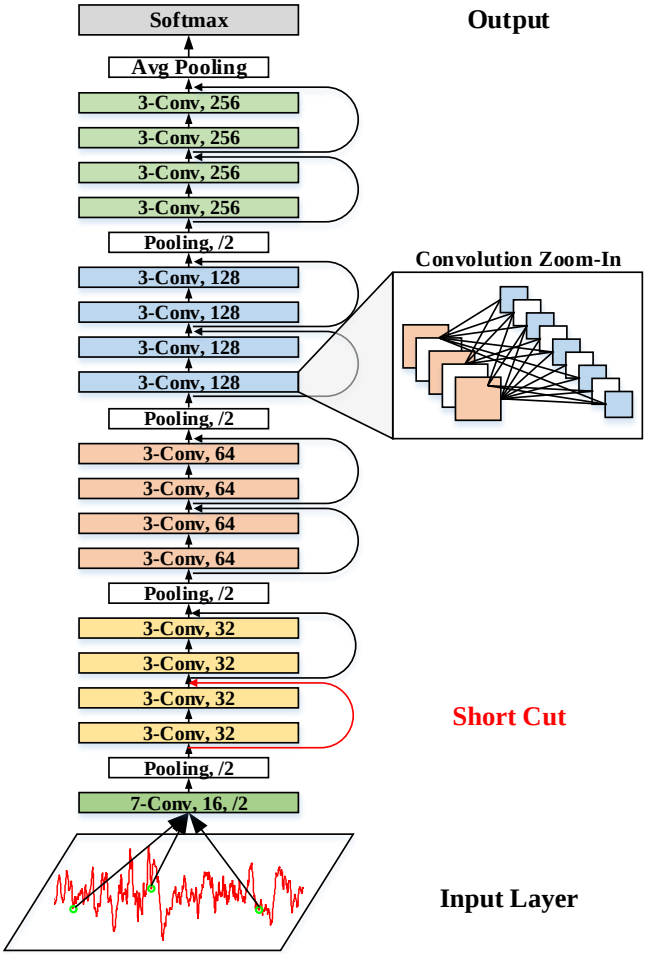

Fig. 2. The framework of our fine-grained analysis module. The whole design is enlightened by the deep residual network.

Architecture: Our fine-grained analysis module is illustrated in Figure 2. The entire network consists of five key parts, a visible input layer, a large-size convolutional layer, residual network blocks, an average pooling and a Softmax output. The visible input layer uses the compressed data $\widehat{y}$ without any feature extraction operations. Then, the input data is processed by the first large-size convolutional layer. *Note that the convolution is 1-D operation in all our fine-grained analysis modules.* The large reception field next to the visible layer can provide enough information to extract the low-level feature, which is critical for the deep layers' processing.

The residual network block is the core part of our fine-grained analysis design. One typical block includes two convolutional layers and uses a small kernel size, such as 3, to learn the data representation on a fine-grained scale. *Note that a batch normalization layer, a dropout layer and a ReLU layer are following each convolutional layer. We omit them here for the simplicity.* In the training phase, the dropout layer is in effect and the dropout ratio is set as 0.5. However, it is closed when the network is in the inference stage. At the end of the residual network block, there is a shortcut to forward its original input to superimpose on the block output feature map as the input of the next layer. In the meantime, the second convolutional layer follows a pooling layer with stride 2. Thus, the next layer will have a double number of convolutional filters but a smaller feature map. We also would like to emphasize that the residual network blocks can be stacked to each other and produce a much deeper network structure. This is because the shortcut can effectively solve the gradient vanishing problem and the dropout/batch normalization can suppress the overfitting when parameter size increases.

After stacking residual network blocks, we also use the global average pooling to replace the fully-connected layer. This technique is more and more popular thanks to its ability to resist the overfitting of the network. Finally, the pooling result is fed into the softmax layer for final prediction retrieval.

Training Phase: Due to the dataset size, we don't train the entire framework from scratch. Instead, we reuse the pre-trained ResNet model to initialize our network. For the dimension difference of the convolutional kernel between our model and ResNet model, we use its diagonal elements to initialize our 1-D kernel. Then we use finetuning to train our designed fine-grained analysis module. Thus, the proposed fine-grained deep neural network can obtain the signal's predominant representation automatically for the classification.

IV. Experiments

In this section, we evaluate the performance of our proposed selective CS architecture in the neural decoding application. We first introduce the experimental setup. The energy and accuracy of proposed selective CS architecture are compared with the state-of-the-art baselines. We also investigate the impact from the confidence score on the trade-off between energy and accuracy. What's more, we compare the accuracy of the entire CS architecture with other popular deep learning backends. Finally, we demonstrate that the direct compression-domain analysis outperforms the analysis on reconstructed signals.

A. Experimental Setup

Datasets: The spike sorting dataset [43] provides us diverse spikes which are from the real-world brain signals. We select 10000 spikes in total from the *difficult-noise-01, difficult-noise-02* cases in the dataset, whose data is challenging to be classified due to the noise contamination in acquisition. Each

spike has the same length of N=64. We randomly choose 8500 spikes as the training set and the rest 1500 spikes as the testing set.

Compressed Sensing: In the compressed sensing setup, we choose four compression ratios CR ($CR = M/N$) in all the experiment to examine the performance of the architectures, i.e., 5%, 10%, 20% and 30%. In the compressed encoding module, we use the inverse discrete wavelet transform (IDWT) as the sparsity-inducing basis Ψ, which can always transform the spike signals into a sparse representation. The Bernoulli random variable is taken as the sensing array. In the quantization module, the uniform quantization strategy is applied. We choose the bitwidth of quantization as a constant number 16, i.e., b=16. For the wireless communication channel for the compressed measurements, the energy model can be defined as $E = C \times M \times b$, where the average energy consumption C = 3 nJ/bit, based on an efficient 350 μW MSK/FSK transmitter [44] specifically designed for medical wearable and implantable devices.

Deep Learning: For the deep learning module in the fine-grained analysis module, we customize the 18-layer deep residual network (ResNet-18) into our spike classification application. We use open-source deep learning framework Caffe [45] to build the ResNet-18 architecture and train it based on our prepared spike waveforms. Instead of training from scratch, we finetune the ResNet-18 with the pre-trained model from Caffe model Zoo [46]. Our batch size is set as 32 and learning rate is 1e-6 with decay rate 0.9 every 4 epochs.

Softmax in Sensor: The softmax model is also trained from our selected spike datasets. Different from random initialization, we use the Softmax layer of deep learning architecture as a good initial estimation. For its hardware implementation, we choose the Synopsys Design Suite [47] to accomplish the design and exploration. We use TSMC 90nm standard cell libraries [48] and implement the design in Verilog with Verilog Compile Simulator (VCS). The design compiler (DC) is adopted to synthesize the Verilog design and Power Compiler is used to report the power consumption.

Baselines: Our first baseline is the conventional CS architecture, which classifies the spikes on the reconstructed signals at the server end. The algorithm is supporting vector machine. Deep belief network (DBN) is another popular deep learning branch. We also choose DBN on compression-domain as the second baseline, which is presented in [8]. The specific setup of DBN structure is two hidden layers with 48 neurons for each. The top classifier is the softmax regression. The activation function is set as "sigmoid", and the batch size is 32 and the learning rate is 0.1.

B. Energy and Accuracy of Selective CS

In this section, we examine the energy consumption and accuracy of the proposed selective CS architecture. We train the deep learning model, ResNet-18, to construct the fine-grained analysis module. We also build the Softmax-based coarse-grained module. The threshold for confidence score is set as 0.4. If the confidence score S is larger than 0.4, the current input is considered as low-effort segment and transmit the final prediction result back to the server end. Otherwise, the wireless transmitter streams the compression measurements $\hat{y}$ back to the server for fine-grained analysis. The screening module and analysis module are both executing the analysis in the compression-domain of the input spike. We use "C-ResNet" to indicate our selective CS architecture, where "C" means compression domain. The two baseline methods, traditional CS architecture and DBN-based CS architecture, are also examined. They are referred as "R-SVM" and "C-DBN", where "R" means reconstruction domain. We collect the information of the three models under all the four compression ratios. The energy consumption is the total value by processing all the spikes in the testing set. The statistics of all the energy consumption is shown in Figure 3 and the comparison of the corresponding classification accuracy is illustrated in Figure 4. Note that the energy is shown under logarithmic scale for a better comparison.

Summary: We can observe from Figure 3 and Figure 4, our proposed ResNet-based selective CS architecture can achieve the best spike classification accuracy and the best energy efficiency among all the architectures. Compared with the traditional CS architecture, our selective CS architecture can achieve more than 60% energy savings under all the compression ratios. It can also improve the spike recognition accuracy over 90%, even under the harsh compression ratio as low as 5%. For the DBN-based solution, although it outperforms the traditional CS architecture, it is still worse than our ResNet-based solution. This demonstrates the effectiveness of selective CS architecture and the importance of the fine-grained deep learning module. It's worth emphasizing that our goal is to improve the energy-efficiency of the front-end sensor node. The fine-grained analysis is not accounted into the total energy consumption because the remote sever is line-powered and not sensitive to the energy.

Energy Comparison: Specifically, compared with the traditional SVM-based CS architecture, our proposed selective architecture gains the energy saving of 59.47%, 62.19%, 72.80% and 71.33% under the compression ratio 5%, 10%, 20% and 30%, respectively. This is because the traditional CS scheme has no in-sensor processing part, so that it suffers from the huge volume of data on the communication channel. When comparing the performance of our ResNet-based solution and the DBN-based solution, our proposed architecture can beat it by a small advantage. This is due to the difference of the Softmax implementation in the coarse-grained screening module. Thanks to training ResNet-18 first, our in-sensor Softmax model obtains a good initialization. However, the DBN case only uses the Gaussian-based random initialization in the Softmax training.

Accuracy Comparison: Our selective CS architecture achieves the best spike classification accuracy, even more than 99% under 30% compression. The accuracy of our proposed scheme outperforms the baselines under all the compression ratios. Under 5% compression ratio, the accuracy of our selective solution is still more than 90%, whereas the traditional CS only obtains 84.24%. However, the improvement of our architecture becomes smaller as the compression ratio

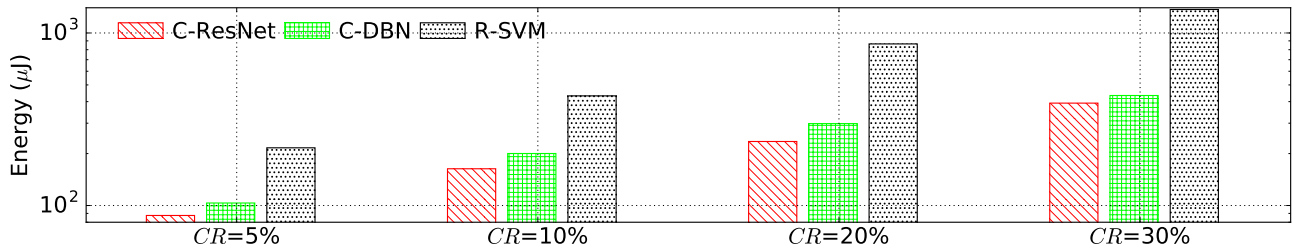

Fig. 3. The energy consumption comparison of the three architectures under the four different compression ratios. For the name of a specific architecture, the hyphen sign connects the signal domain and fine-grained analysis method. "C" indicates the compression domain and "R" is the reconstruction domain.

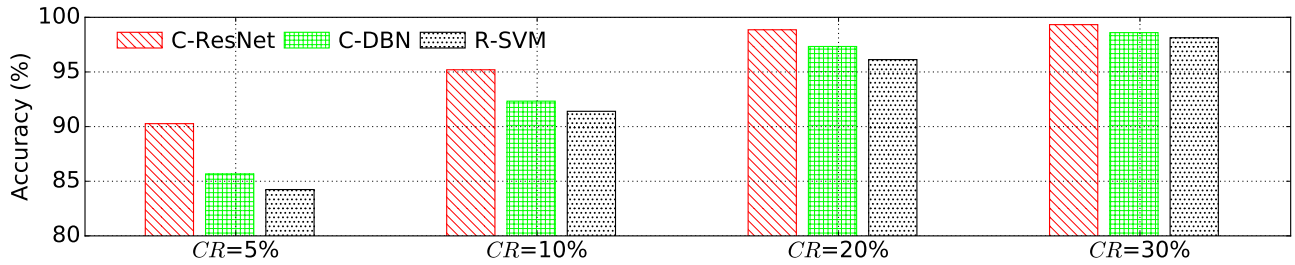

Fig. 4. The comparison of the spike classification accuracy of the three architectures under the four different compression ratios.

increases. The server end can acquire more information under the larger compression ratio. With enough information, the different fine-grained algorithms can have a similar spike classification accuracy.

C. Confidence Score Impact in Selective CS

In this section, we investigate the impact from the confidence score S on energy and accuracy of our proposed selective architecture. The configuration of our ResNet-based scheme is the same as Section IV-B. The compression ratio of compressed sensing module is chosen as 10%. In this experiment, we consider the confidence threshold th_S as a parameter. We range confidence score threshold th_S from 0.0 to 1.0 with step as 0.02. We collect the energy consumption and classification accuracy of our selective CS architecture under each different th_S. The final statistical result is shown in Figure 5.

We can observe that as the confidence score threshold th_S increase, the accuracy of proposed selective architecture is improved. In the meantime, the energy budget of the sensor end is also increasing rapidly. In the beginning, the threshold th_S is 0, which means all coarse-grained prediction result is the confident low-effort task. So the transmitter only sends the prediction results back to the server end. This aggressively decreases the energy consumption on the wireless channel due to the data volume reduction. Thus, the spike classification accuracy is equal to the discrimination result of the in-sensor Softmax implementation.

As the confidence threshold increases, our selective architecture obtains a rapid increasing period. In this threshold slot, the accuracy increases rapidly while only consuming a reasonable amount of the energy. Considering the trade-off between accuracy and energy budget, this stage is good for the energy-efficient design requirement. The reason of forming this period is that as the threshold increases, more high-effort input spikes are transmitted back to the server end for fine-grained analysis. These spikes can confuse the coarse-grained Softmax screening, but can be recognized by high-discrimination deep learning model.

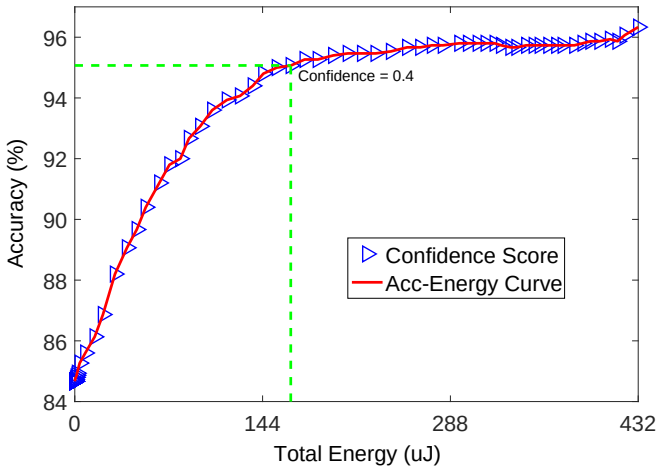

Fig. 5. The accuracy-energy curve under different confidence thresholds.

Finally, the accuracy approximates to saturation but the energy budget is still increasing. This stage is just wasting the energy without significant accuracy improvement. This is because as the confidence threshold continuously increases, more low-effort spikes are not screened by the coarse-grained screening module. Their compression representations are transmitted back to the fine-grained analysis. Although they obtain the correct result eventually, they waste the energy consumed on the wireless channel. In our experiment, this turning point happens around th_S is equal to 0.4. This is why we choose the confidence score threshold as 0.4 in Section

IV-B. Therefore, we can find that the confidence threshold can affect the accuracy and energy budget of our selective CS architecture simultaneously. We can find a good trade-off between energy and accuracy of selective architecture by selecting an appropriate confidence threshold.

D. Deep Learning Model of Selective CS Architecture

The deep learning model in the fine-grained analysis is a significant factor affecting the accuracy and energy consumption of the selective CS architecture. Its impact on energy is due to its contribution to the initialization of the in-sensor Softmax model. In this experiment, we explore the architecture performance under different popular deep learning models. We choose AlexNet, VGGNet and GoogLenet to design the fine-grained analysis module at the server end. All the convolutional layers in these networks are transformed into 1D convolution. We keep their architecture details and training scheme. They are all trained using finetuning and initialized by their pre-trained models from Caffe model zoo. We also choose two compression ratio, 10% and 30%, to simulate the low and high compression representations. We collect the spike classification accuracy in all the cases. The final statistics are listed in TABLE I.

TABLE I
ACCURACY OF SELECTIVE CS ARCHITECTURE USING DIFFERENT DEEP LEARNING MODELS.

CR	**C-AlexNet**	**C-VGGNet**	**C-GoogLenet**
10%	89.40%	93.87%	94.33%
30%	96.33%	98.80%	99.00%
CR	**C-ResNet**	**C-DBN**	**R-SVM**
10%	**95.20%**	92.33%	91.40%
30%	**99.33%**	98.60%	98.19%

We can see that our ResNet-based solution achieves the best accuracy under the both low and high compression ratios. From network architectures, the capability of deep models is ordered as ResNet, GoogLenet, VGGNet and AlexNet. This conclusion is confirmed by the final accuracy. Their accuracy decreases according to this order. Due to the dataset size, we choose ResNet-18 as the base deep learning model. The deeper ResNet architecture is also promising to achieve better accuracy. On the other hand, if we compare the performance of these deep models with DBN and SVM, we can find that the AlexNet behaves the worst. This is because AlexNet has deeper network architecture than DBN and SVM, but it doesn't apply the advanced training skills to resist over-fitting problem [49], [50]. Therefore, we can conclude that the ResNet-based selective CS architecture can achieve the best performance among all the state-of-the-art machine learning models.

E. Compression Domain and Reconstruction Domain

In all above experiments, we directly use compression-domain data to feed the deep learning models for accurate predictions because it can preserve the salient information of the high-dimensional representation. Although this is supported by mathematicians [26], we would like to confirm this assumption in this experiment. We only compare the spike classification accuracy of different models by feeding data on reconstruction domain. This is done by omitting the in-sensor coarse-grained processing module. We choose ResNet-18, DBN and SVM as the benchmark. We adopt CVX tool to execute the reconstruction from the compressed measurements. We also collect the spike classification accuracy of all these three models on both compression domain ("C") and reconstruction domain ("R"). All the information is summarized in Figure 6.

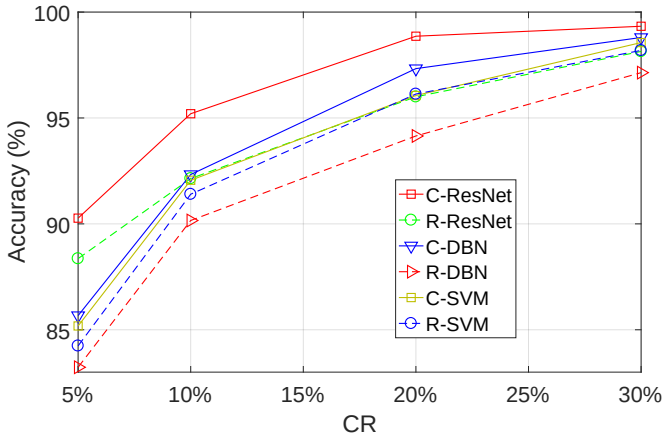

Fig. 6. The accuracy comparison of fine-grained analysis methods on both compression and reconstruction domains.

We can observe from Figure 6 that the compression domain achieves better accuracy in all the cases compared with the reconstruction domain by 1% to 3%. In the figure, we use solid lines to indicate the accuracy trend in compression domain and dashed lines to show the accuracy in reconstruction domain. In the low CR case, the accuracy difference is much larger than that in the high CR case. This is because high CR case can always provide more information for signal reconstruction in order to obtain more accurate results. As the distortion of the spikes decreases, the different models tend to achieve a similar accuracy result. One interesting finding is that although ResNet in compression domain has the best accuracy, the DBN in compression domain can beat the ResNet in reconstruction domain. This demonstrates that the input data quality is more significant than the network architecture for our selective CS framework. Another interesting finding is that SVM behaves well in the reconstruction domain. It is better than the DBN case and slightly worse than our ResNet case. This demonstrates that the shallow learning has better noise-resistant ability than the deep learning model.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a selective CS architecture for wireless implantable neural decoding. We designed two cooperative modules, a coarse-grained screening module designed by softmax regression and a fine-grained analysis module equipped with customized deep residual neural network, for different effort-level classification tasks. The screening module processed the low-effort classification and transmitted the high-effort task to the fine-grained module at the server

end. Experimental results validated the data analysis in the compressed domain. By the collaboration of the two proposed modules, our selective CS architecture gained more than 60% energy savings, while even improving the spike classification accuracy when compared to the traditional CS architecture.

In the future, we plan to deeply combine the signal analysis procedure with the compression operations of compressed sensing framework and deploy the architecture in real world. Another promising direction is to design a controller which can adjust the configuration of the compressed sensing functional modules and the confidence threshold in the coarse-grained screening module to improve the energy-efficiency of the entire system further.

ACKNOWLEDGMENT

This work is in part supported by the U.S. National Science Foundation grants under CNS-1423061, ECCS-1462498, the National Science Foundation of Chine (61772471), and International Science & Technology Cooperation and Exchanges Plan in ShaanXi Province of China under Grant Number 2016KW-044.

REFERENCES

[1] M. Aghagolzadeh and K. Oweiss, "Compressed and distributed sensing of neuronal activity for real time spike train decoding," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 2, pp. 116–127, 2009.

[2] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.

[3] K. D. Wise, D. Anderson, J. Hetke, D. Kipke, and K. Najafi, "Wireless implantable microsystems: high-density electronic interfaces to the nervous system," *Proceedings of the IEEE*, vol. 92, no. 1, pp. 76–97, 2004.

[4] F. Chen, A. P. Chandrakasan, and V. Stojanović, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*. IEEE, 2010, pp. 1–4.

[5] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[6] C. Song, A. Wang, F. Lin, R. Zhao, Z. Jin, and W. Xu, "A tempo-spatial compressed sensing architecture for efficient high-throughput information acquisition in organs-on-a-chip," in *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*. IEEE, 2017, pp. 229–232.

[7] A. Wang, C. Song, and W. Xu, "A configurable quantized compressed sensing architecture for low-power tele-monitoring," in *Green Computing Conference (IGCC), 2014 International*. IEEE, 2014, pp. 1–10.

[8] A. Wang, C. Song, X. Xu, F. Lin, Z. Jin, and W. Xu, "Selective and compressive sensing for energy-efficient implantable neural decoding," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[9] J. Zhang, Y. Suo, S. Mitra, S. P. Chin, S. Hsiao, R. F. Yazicioglu, T. D. Tran, and R. Etienne-Cummings, "An efficient and compact compressed sensing microsystem for implantable neural recordings," *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 4, pp. 485–496, 2014.

[10] Y. Suo, J. Zhang, T. Xiong, P. S. Chin, R. Etienne-Cummings, and T. D. Tran, "Energy-efficient multi-mode compressed sensing system for implantable neural recordings," *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 5, pp. 0–0, 2014.

[11] R. Fallahzadeh, J. P. Ortiz, and H. Ghasemzadeh, "Adaptive compressed sensing at the fingertip of internet-of-things sensors: An ultra-low power activity recognition," in *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 996–1001.

[12] F. Lin, C. Song, X. Xu, L. Cavuoto, and W. Xu, "Sensing from the bottom: Smart insole enabled patient handling activity recognition through manifold learning," in *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*. IEEE, 2016, pp. 254–263.

[13] A. Wang, L. Chen, and W. Xu, "Xpro: A cross-end processing architecture for data analytics in wearables," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*. ACM, 2017, pp. 69–80.

[14] M. Sugano, R. Isaksson, Y. Nakajima, and H. Yanagihara, "Shot genre classification using compressed audio-visual features," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2. IEEE, 2003, pp. II–17.

[15] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 245–250.

[16] H. Reboredo, F. Renna, R. Calderbank, and M. R. Rodrigues, "Compressive classification," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 674–678.

[17] M. Shoaib, N. K. Jha, and N. Verma, "Signal processing with direct computations on compressively sensed data," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 1, pp. 30–43, 2015.

[18] R. Braojos, H. Mamaghanian, A. D. Junior, G. Ansaloni, D. Atienza, F. J. Rincón, and S. Murali, "Ultra-low power design of wearable cardiac monitoring systems," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, p. bbw068, 2016.

[21] A. Wang, Z. Jin, C. Song, and W. Xu, "Adaptive compressed sensing architecture in wireless brain-computer interface," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 173.

[22] A. M. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ecg and emg wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[23] A. Wang, W. Xu, Z. Jin, and F. Gong, "Quantization effects in an analog-to-information front end in eeg telemonitoring," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 2, pp. 104–108, 2015.

[24] P. K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*. IEEE, 2009, pp. 144–148.

[25] X. Wang, Z. Zhao, Y. Xia, and H. Zhang, "Compressed sensing for efficient random routing in multi-hop wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 7, no. 3-4, pp. 275–292, 2011.

[26] B. Coppa, R. Héliot, D. David, and O. Michel, "Classification from compressive representations of data," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1454–1458.

[27] M. Shoaib, N. K. Jha, and N. Verma, "A compressed-domain processor for seizure detection to simultaneously reduce computation and communication energy," in *Custom Integrated Circuits Conference (CICC), 2012 IEEE*. IEEE, 2012, pp. 1–4.

[28] J. Ko, K. Klues, C. Richter, W. Hofer, B. Kusy, M. Bruenig, T. Schmid, Q. Wang, P. Dutta, and A. Terzis, "Low power or high performance? a tradeoff whose time has come (and nearly gone)." in *EWSN*. Springer, 2012, pp. 98–114.

[29] V. Handziski, J. Polastre, J.-H. Hauer, and C. Sharp, "Flexible hardware abstraction of the ti msp430 microcontroller in tinyos," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 277–278.

[30] V. Karkare, S. Gibson, and D. Marković, "A 75-μw, 16-channel neural spike-sorting processor with unsupervised clustering," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 9, pp. 2230–2238, 2013.

[31] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[32] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views

of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
[33] H. Li, M. R. Min, Y. Ge, and A. Kadav, "A context-aware attention network for interactive question answering," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 927–935.
[34] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[41] V. S. Huang, R. Shadmehr, and J. Diedrichsen, "Active learning: learning a motor skill without a coach," *Journal of neurophysiology*, vol. 100, no. 2, pp. 879–887, 2008.
[42] R. Andraka, "A survey of cordic algorithms for fpga based computers," in *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*. ACM, 1998, pp. 191–200.
[43] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.
[44] J. L. Bohorquez, A. P. Chandrakasan, and J. L. Dawson, "A 350 w cmos msk transmitter and 400 w ook super-regenerative receiver for medical implant communications," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 4, pp. 1248–1259, 2009.
[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.
[46] Y. Jia, "Bvlc caffe model zoo," 2013.
[47] H. Bhatnagar, *Advanced ASIC Chip Synthesis: Using Synopsys® Design CompilerTM Physical CompilerTM and PrimeTime®*. Springer Science & Business Media, 2007.
[48] M. Sanie, M. Côté, P. Hurat, and V. Malhotra, "Practical application of full-feature alternating phase-shifting technology for a phase-aware standard-cell design flow," in *Design Automation Conference, 2001. Proceedings*. IEEE, 2001, pp. 93–96.
[49] P. Ballester and R. M. de Araújo, "On the performance of googlenet and alexnet applied to sketches." in *AAAI*, 2016, pp. 1124–1128.
[50] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sensing*, vol. 9, no. 8, p. 848, 2017.

Chen Song (S'14) received the B.S. degree in optic science and engineering from Fudan University, and the M.S. degree in electrical engineering from the State University of New York at Buffalo, where he is currently pursuing the Ph.D. degree with the Department of Computer Science, under the direction of Prof. W. Xu. His current research focuses on Mobile Smart Health, Cyber Physical Security and Emerging Biometrics.

Aosen Wang (S'15) is a third-year Ph.D. student of Computer Science and Engineering at University at Buffalo, State University of New York (SUNY). He received his B.S. degree in Electrical Engineering from the University of Science and Technology of China (USTC) in 2011. Then he joined Vimicro as an algorithm and software engineer. His research interests include low-power computer architecture and energy-efficient machine learning.

Feng Lin (S'11-M'15) received the B.S. degree from Zhejiang University, China, the M.S. degree from Shanghai University, China, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Tennessee Technological University, USA. Currently, he is an Assistant Professor with the Department of Computer Science and Engineering, University of Colorado Denver, USA. Prior to that, he was a Research Assistant Professor at the State University of New York (SUNY) at Buffalo, USA, and worked for Alcatel-Lucent (now Nokia). He won the First Prize Design Award in 2016 International 3D printing competition and best paper award in 2017 IEEE BHI conference. His research interests lie in areas of mobile sensing, smart health and cyber-physical security.

Jian Xiao received the B.Sc. degree in electronic engineering from the Chengdu University of Technology, China, in 1997 and the Ph.D. degree from the Lanzhou University, China, in 2008. He became a Research Assistant in the Changan University in June 2008. He is currently an Associate Professor in the Changan University. His research interests include signal processing, artificial intelligence applications, pattern recognition, and computer vision.

Xinwei Yao is an Associate Professor with the College of Computer Science and Technology at the Zhejiang University of Technology (ZJUT), Hangzhou, China. He received the B.S. in Mechanical and Electrical Engineering and Ph.D. degree in Information Engineering from ZJUT, in 2013. From March 2012 to February 2013, he was a visiting scholar at the Loughborough University, Leicestershire, UK. From August 2015 to July 2016, he was a visiting professor at the University of Buffalo, The State University of New York, Buffalo, NY, USA. He was the recipient of the Outstanding Doctoral Thesis Award and the Distinguished Associate Professor Award from ZJUT in 2013 and 2014. He has served on technical program committees of many IEEE/ACM conferences. He is a Member of IEEE and ACM. His current research interests are in the area of Terahertz-Band Communication Networks, Electromagnetic Nanonetworks, Wireless Ad Hoc and Sensor networks, Wireless Power Transfer and the Internet of Things.

Wenyao Xu (M'13) received the Ph.D. degree from the Electrical Engineering Department, University of California, Los Angeles, CA, USA, in 2013. Currently, he is an Assistant Professor of the Computer Science and Engineering Department, the State University of New York (SUNY) at Buffalo, Buffalo, NY, USA. His current research focuses on system technologies and their applications on Wireless Health Care, Internet of Things and Cyber-Physical Systems. He owned five licensed U.S. and international patents and has authored more than 100 peer-reviewed journal and conference papers. Dr. Xu received the Best Paper Award of IEEE International Conference on Biomedical and Health Informatics (BHI) in 2017 and the 1st Place Best Design Award of International 3D Printing Competition in 2016, the Best Paper Award of IEEE Conference on Implantable and Wearable Body Sensor Networks (BSN) in 2013, and the Best Demonstration Award of ACM Wireless Health Conference (WH) in 2011.