

A Low-Energy Machine-Learning Classifier Based on Clocked Comparators for Direct Inference on Analog Sensors

Zhuo Wang, *Member, IEEE*, and Naveen Verma, *Member, IEEE*

Abstract—This paper presents a system, where clocked comparators consuming only CV^2 energy directly derive classification decisions from analog sensor signals, thereby replacing instrumentation amplifiers, ADCs, and digital MACs, as typically required. A machine-learning algorithm for training the classifier is presented, which enables circuit non-idealities as well as severe energy/area scaling in analog circuits to be overcome. Furthermore, a noise model of the system is presented and experimentally verified, providing a means to predict and optimize classification error probability in a given application. The noise model shows that superior noise efficiency is achieved by the comparator-based system compared with a system based on linear low-noise amplifiers. A prototype in 130-nm CMOS performs image recognition of handwritten numerical digits, by taking raw analog pixels as the inputs. Due to pin limitations on the chip, the images with $28 \times 28 = 784$ pixels are resized and downsampled to give 47 pixel features, yielding an accuracy of 90% for an ideal ten-way classification system (MATLAB simulated). The prototype comparator-based system achieves equivalent performance with a total energy of 543 pJ per ten-way classification at a rate up to 1.3 M images per second, representing $33\times$ lower energy than an ADC/digital-MAC system.

Index Terms—Classification, comparators, low-energy accelerator.

I. INTRODUCTION

THE Internet of Things (IoT) represents a compelling vision in which intelligent sensing is a critical requirement. With over a trillion devices being able to gather and exchange relevant data, it will be essential that the data be reduced to aggregated, high-value information up front, both to manage communication energy and total communication bandwidth available for the devices, but also to enable efficient processing and control by the centralized systems they interact with. A challenge in creating intelligent sensing devices capable of extracting such information from embedded signals, is that the signals are typically derived from complex physics of

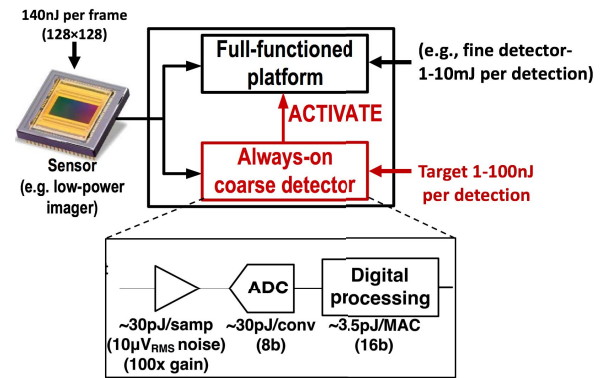


Fig. 1. Architecture of image sensing IoT node, employing low-energy, always-on detector to trigger full-functioned platform. Rough energy numbers are obtained directly or through derivation from numbers previously reported for image sensors [4], digital classification platforms [1]–[3], and analog frontends [5], [6].

the real world with dependence on unpredictable parameters. This substantially limits the viability of analytical models for extracting inferences from the signals, and has made data-driven methods based on machine learning of paramount importance.

A key aspect of intelligent sensing is *always-on sensing and inference*, so that devices can respond as physical event of interest occur. Machine-learning algorithms have demonstrated tremendous success for extracting accurate inferences, especially from vision data, a particularly important IoT sensing modality. The problem is that state-of-the-art machine-learning models can be complex, requiring several milli-Joules of energy per classification [1]–[3], precluding always-on sensing and inference. To address this, the high-level node architecture shown in Figure 1 is being adopted. Here, a low-energy subsystem provides somewhat coarser, but always-on, sensing and inference. This can then selectively activate a full-functioned platform, which possibly provides higher-accuracy inference along with other energy-intensive capabilities, but with reduced duty cycle. The low-energy subsystem must still employ machine-learning models, in order to address the complexities of embedded real-world signals; however, it must do so at orders-of-magnitude lower energy.

Energy analysis of the various functions (and associated circuit blocks) required in a conventional digital implementation of the always-on subsystem (rough numbers in Figure 1) shows that instrumentation amplifiers, ADCs,

Manuscript received October 28, 2016; revised April 3, 2017; accepted May 9, 2017. Date of publication June 1, 2017; date of current version October 24, 2017. The work of Z. Wang was supported by the Honorable Fellowship from Princeton University. This work was supported in part by AFOSR under Grant FA9550-14-1-0293, in part by NSF under Grant CCF-1253670, and in part by Systems on Nanoscale Information fabriCs, one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. This paper was recommended by Associate Editor X. Liu. (Corresponding author: Zhuo Wang.)

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: pku.zwang@gmail.com; nverma@princeton.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2017.2703880

1549-8328 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

and digital processors (e.g., performing MAC operations) all contribute significantly to energy. To address this, this paper presents a mixed-signal machine-learning classifier that replaces all of these, directly taking analog sensor data as inputs and providing classification decisions as outputs. Mixed-signal implementation raises numerous challenges for accurate classification, in terms of circuit non-idealities and dynamic-range requirements. To address these without increasing system energy, a low-complexity classifier-training algorithm is presented. This can be implemented directly on a memory-limited embedded processor, presumed within the full-functioned subsystem of Figure 1a. The prototype classification system is demonstrated in an image-recognition application (detecting handwritten numerical digits from the MNIST [7] dataset), using raw analog pixel voltages as inputs and providing 10-way digit classification at the same accuracy level of an ideal ADC/digital-MAC system. While there exists several prior works targeting analog or mixed-signal machine learning classifier implementations [6], [8]–[13], we demonstrate here the implementation of an analog classification system that directly takes analog sensor inputs for inference through a comparator structure. It achieves substantial energy savings and same classification accuracy compared to the corresponding digital implementation. The energy per MAC operation (51fJ) is the best ever reported. The specific contributions of this work are as follows:

1. We present a binary classification circuit based on a clocked-comparator topology that directly takes analog sensor signals, replacing front-end instrumentation amplifiers, ADCs, and digital MAC units. Only CV^2 energy is consumed, yielding $33\times$ energy savings compared to an optimized ADC/digital-MAC implementation in an image-recognition application demonstrated.
2. We present a machine-learning algorithm based on boosted linear classifiers, which specifically overcomes analog circuit non-idealities and enables low precision of the linear-classifier weights to avoid severe energy/area scaling in an analog implementation. With the proposed algorithm, the demonstrated system achieves classification accuracy with just 4-b weights at the level of an ideal ADC/digital-MAC system requiring at least 10-b weights.
3. We develop a noise model of the system, perform noise analysis, and experimentally verify the noise model, demonstrating that the proposed circuit implementation presents a superior noise-power tradeoff compared to linear low-noise amplifiers conventionally used (as previously seen for conventional comparator structures [14]). This suggests that the mixed-signal implementation is efficient at the system level not only from an energy-per-operation perspective, but also from a noise perspective.

The remainder of this paper is organized as follows. Section II presents background both on existing classification systems targeting resource-constraint platforms, as well as on algorithmic design principles employed within the proposed system to overcome circuit non-idealities. Section III presents the implementation details of the clocked-comparator classifier circuit. Section IV presents the training algorithm

for overcoming analog non-idealities as well as computation-precision constraints. Section V derives the noise model for the classification system and presents noise analysis for the MNIST image-recognition application demonstrated. Section VI presents the prototype and measurement results, along with demonstration of a handwritten digit-detection application. Finally, Section VII concludes.

II. BACKGROUND

The need for energy efficiency in various signal-processing and inference systems has renewed a focus on analog computation. This has been driven both by the interest in applying embedded computation to sensor signals, which are typically analog, and by the notion that analog computation, for the reduced precision often tolerable in sensing applications, can be more energy efficient than digital computation by taking advantage of the richer physics possible in circuit networks than simply boolean switching [15]. The sections below provide background on the state of the art in mixed-signal computation and then on recently reported machine-learning algorithms for enhancing the precision achievable in the presence of implementation non-idealities, typically arising with analog computation.

A. Mixed-Signal MAC Accelerators

For embedded signal-processing and inference computations, structures for implementing MAC operations have been of greatest focus. It has long been known that an efficient method of implementing multiplication via analog circuits has been in the charge domain, through switched-capacitor architectures. While traditional architectures have made use of op-amp circuits to ensure complete charge transfer and minimal sensitivity to parasitics, more recently passive architectures have emerged. While these suffer from various non-idealities, resulting errors have either been tolerable within applications (e.g., causing only gain error) or mitigated through careful design to minimize the effects of parasitics. This has resulted in demonstrations of accelerators for FFT [16], general matrix multiplication [17], and classification [6], reporting $12\text{--}29\times$ energy improvement over digital implementations. The designs have also shown that such architectures can benefit from implementation in deeply scaled technologies, with increasingly advanced nodes being employed and yielding corresponding improvements to energy efficiency. More recently analog architectures have also emerged that not only address computational energy, but also memory-accessing/communication energy. For instance, by implementing computation using the bit cells of an SRAM, [10] achieves highly-parallel operation while avoiding energy expenditure to explicitly move the stored data to a point of computation external to the SRAM. Thus, analog implementation has resulted in increased energy efficiency of computation as well as alternate architectural options whereby communication energy is significantly reduced.

However, a key tenet of analog computation has been that it is most beneficial for low-precision computation. This is true practically, because analog architectures are more sensitive to

device mismatch, charge-injection errors, coupling noise, etc. But it is also true fundamentally, because analog computation has a dynamic range that is limited by the voltage/current headroom and noise level of a circuit, both of which are energy intensive to address [15]. This is especially problematic for multiplication operations, since these tend to substantially increase the dynamic-range requirements. For this reason, all of the architectures cited above restrict themselves to low computational precision and demonstrate applications where such precision is acceptable. One exception is [6], which employs mixed-signal floating-point multiplication within an ADC, where a 4-bit mantissa restricted to a value less than 2 is applied in the analog domain, and an exponential multiplier with base 2 is applied in the digital domain (via barrel shifting). In this way, the analog dynamic range required is only increased by a factor of 2, even though multipliers of much larger value can be applied.

The approach taken in this work aims to address the reduced precision of analog computation on multiple levels, differently than the approaches above. First, it exploits a machine-learning algorithm previously reported (described below) to address non-idealities in analog circuits (mismatch, charge injections, etc.). Second, it restricts the precision of specific variables, namely weights corresponding to the machine-learning model, but employs a specialized algorithm referred to as Constraint Resolution Regression (CRR), for achieving optimal learning with the reduced-precision weights. Third, it employs a topology wherein circuit noise affecting the dynamic range supported for analog inputs is substantially reduced compared to typical circuits based on linear settling.

B. Data-Driven Hardware Resilience (DDHR)

A tolerance to errors can potentially open up many new architectural choices for substantially reducing system energy. While it has been argued that many sensor-inference applications inherently exhibit some tolerance to errors [18], several works have shown this tolerance to be limited, restricting the architectures possible [19], [20] or restricting how aggressively circuit approaches for energy reduction can be applied [21]. On the other hand, DDHR is an approach that aims to substantially increase tolerance to errors beyond the inherent level, by taking advantage of data-driven training of an inference model enabled by machine-learning algorithms [22].

Figure 2a illustrates the concept of DDHR, in the case of a classification system based on supervised learning. In the system, it is assumed that the feature-extraction stage preceding the classifiers is affected by non-idealities. However, feature-computation errors arising as a result are viewed simply as perturbing the data distributions corresponding to the classes of data. Employing data-driven training, explicitly with the error-affected feature data, allows a classification model to be learned for these new distributions. The resulting model is referred to as an *error-aware model*. This is illustrated in Figure 2b, where the left-hand plot shows the distribution of features (actually derived from a hardware experiment) for a fault-free system, and the right-hand plot shows the distribution of features for a system in which

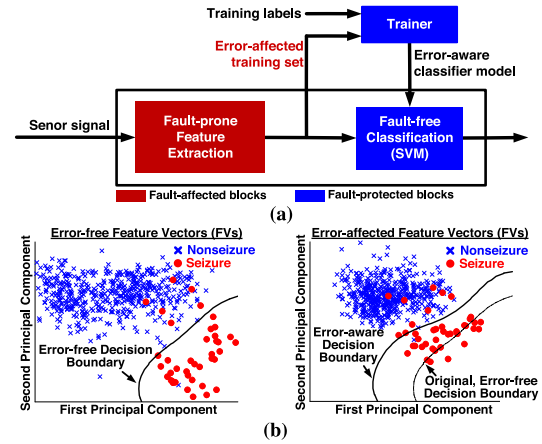


Fig. 2. Illustration of DDHR [22], showing (a) its use in a supervised-learning classification system, and (b) the manner in which an error-aware model is formed and enhances accuracy in the presence of errors (principal component analysis is used to project features data to two dimensions to aid visualization).

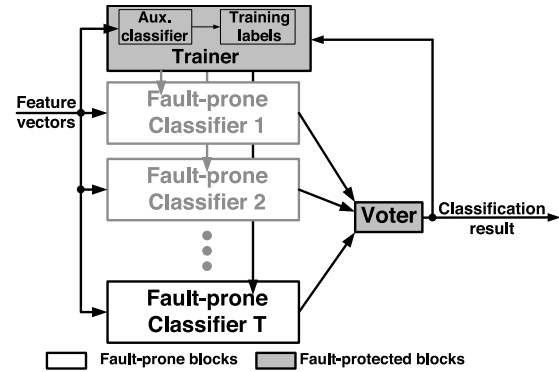


Fig. 3. Illustration of EACB system architecture [23].

randomized stuck-at-1/0 faults have been introduced [22]. As seen, a classification decision boundary trained to data from the fault-free system is sub-optimal for the fault-affected system, but a decision boundary trained using the fault-affected data substantially restores accurate classification.

1) *Error-Adaptive Classifier Boosting (EACB)*: In the system of Figure 2a, it appears that only faults in the stages preceding the classifier can be handled (i.e., feature extractor), such so that the error-aware model is reliably applied. In fact, the idea of DDHR has been extended to enable substantial resilience to faults in the classifier as well. This is done through an algorithm referred to as EACB [23]. EACB leverages a machine-learning algorithm known as adaptive boosting (AdaBoost), which uses an ensemble of weak classifiers to form a strong classifier [24] (in machine learning, a strong classifier is one that can be trained to fit arbitrary data distributions, while a weak classifier is one that cannot be). In AdaBoost, the weak classifiers are trained iteratively in a way that is biased by the fitting errors from the previous iteration. EACB, illustrated in Figure 3, extends this approach by using the specific instances of weak-classifier implementations (affected by non-idealities) to bias the iterations of

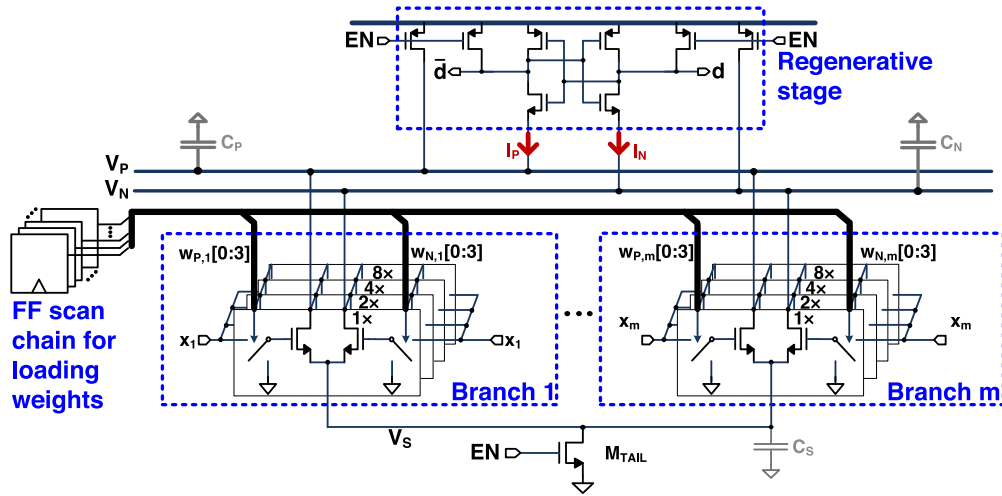


Fig. 4. Comparator-based binary linear classifier employs configurable NFET branches.

weak-classifier training. As a result, weak-classifier models are adaptively learned to overcome not only fitting errors but also errors resulting from non-ideal implementation. The theory of AdaBoost notes that a strong classifier can be achieved even with extremely weak classifiers [24], making it possible for EACB to overcome severe non-idealities in the weak-classifier implementations [23].

2) *Training the Error-Aware Model*: Given that hardware non-idealities can be a result of or subject to variations, a different error-aware model may be required for each instance of a system. Thus, previous work on DDHR and EACB has also explored how systems can self-train their own error-aware models [22], [23]. This raises two concerns: (1) availability of a training set; and (2) computational resources required for training. In terms of the training set, supervised learning requires training data and corresponding training labels. While error-affected training data is readily available from within the system, labels are typically provided from an external source. However, in an architecture such as Figure 1a, a high-performance classifier free from non-idealities can be implemented on the full-functioned node, and during infrequent training phases can provide class declarations used by a trainer to *estimate* the labels. Previous work [22] shows that this enables a level of performance limited by the accuracy of the high-performance system or that of a non-ideal system trained using perfect labels. In terms of computational resources, in many scenarios the energy and latency of training can be tolerated. This is because training typically occurs infrequently and not in real time; thus its energy can be amortized and its latency does not strongly affect system operation. However, one aspect that is of concern is the amount of embedded memory required for the training set, in order to ensure low generalization error (i.e., avoid overfitting). In [23], a training algorithm is proposed for EACB, which takes advantage of the fact that weak classifiers (e.g., linear classifiers) exhibit reduced susceptibility to overfitting. This allows the training set for each iteration to be substantially reduced, thereby reducing the instantaneous memory requirements; but, by acquiring a new training set at each iteration, the training set

diversity is enhanced for the strong ensemble. Thus training on a full functioned node (as in Figure 1a) can be achieved.

III. COMPARATOR-BASED CLASSIFIER

This section describes the proposed implementation of a weak binary classifier based on a clocked-comparator structure. As described in Section IV, a strong binary classifier is formed by combining several such classifiers via the EACB algorithm, and, as described in Section VI, a multi-class classifier is formed by employing multiple strong binary classifiers.

The proposed clocked-comparator structure implements a linear classifier, which corresponds to taking the inner product between a feature vector \vec{x} to be classified and a weight vector \vec{w} (derived from training), and then performing sign thresholding: $\text{sign}(\vec{w} \cdot \vec{x})$. Within a system, it is assumed that the elements of \vec{x} correspond to signals from different sensor-input channels. Figure 4 shows the circuit proposed to implement this. It consists of m configurable branches, driven by the analog sensor-input signals. In the prototype $m = 48$. Each branch consists of two sets of NFETs having binary-scaled widths (i.e., $1\times$, $2\times$, $4\times$, $8\times$, ...), whose gate voltages are digitally configurable to ground or the analog sensor input. Treating the currents from the two sets of NFETs as a differential signal, the total branch current corresponds to signed multiplication between the analog sensor input and a value corresponding to the gate configurations. The sets of NFETs from all branches are then connected to the positive and negative summing nodes (V_P/V_N), respectively, for current accumulation. Thus, an inner-product computation is implemented, where \vec{x} corresponds to the analog sensor inputs and \vec{w} corresponds to the gate configurations. With V_P/V_N previously pre-charged, raising EN causes the accumulated branch currents to discharge the node capacitances C_P/C_N , which triggers comparison by a regenerative stage, thus implementing classification. While the analog sensor inputs are assumed to be continuous-time signals, the gate configurations are loaded via a shift register once after classifier training.

Details of the operation are shown in Figure 5a. When EN is low, both positive and negative summing nodes V_P/V_N are

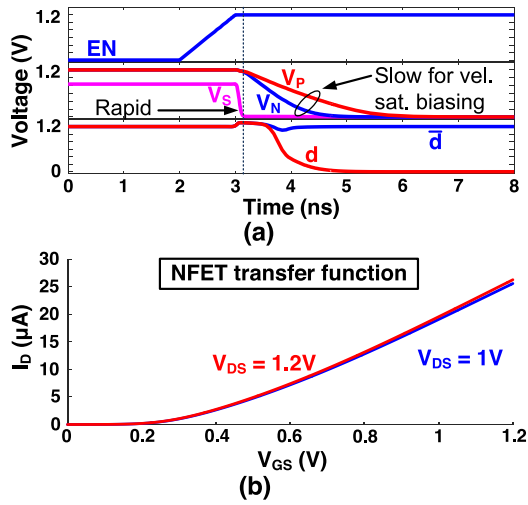


Fig. 5. Operation showing that (a) tail device M_{TAIL} and branch NFETs are sized for velocity-saturation biasing in the range of interest (before regeneration), and (b) this yields roughly linear relationship between the analog sensor-input voltage (NFET V_{GS}) and the branch current (NFET I_D).

pre-charged to V_{DD} . Following assertion of EN , the source node V_S is pulled down rapidly, by appropriately sizing the tail device M_{TAIL} . On the other hand, V_P/V_N are pulled down relatively slowly by the aggregate branch currents I_P/I_N , due to sizing of the branch NFETs. Thus, the branch NFETs are biased to be primarily in velocity saturation. This causes nearly linear relationship between the input voltages and branch currents, as shown in Figure 5b (for V_{DS} of 1.2V and 1.0V, below which regeneration is triggered). With the currents from all 48 branches adding together, the structure achieves the 48 MAC operations and sign thresholding required in a linear classifier all in the analog domain, with an energy of just

$$E_{CLASS} \approx C_P V_{DD}^2 + C_N V_{DD}^2 + C_S (V_{DD} - V_{t,n}) V_{DD} \quad (1)$$

(i.e., $C_P/C_N/C_S$ are the dominant capacitances). This is much lower than the energy expected for a digital implementation, and the throughput is high, because all 48 MAC operations are performed in parallel in the time required for a single regenerative stage.

Although the analog implementation proposed leads to significant energy and throughput advantages, it faces several challenges. First, analog implementation suffers from numerous non-idealities, such as variations, charge-injection errors, transconductance non-linearity, etc. Second, the digital weights obtained from training cause exponential scaling of energy and area in an analog implementation, by setting the number of binary-scaled branch NFETs. For instance, a unit NFET corresponding to an LSB of the weights has a W/L of $0.2\mu m/1\mu m$, and a capacitance contribution to C_P/C_N of roughly 12fF (including wiring capacitances). With the exponential scaling involved, we estimate the area of a differential NFET branch to exceed that of a digital multiplier at an input resolution of 15b and the energy to exceed that of a digital multiplier at an input resolution of 14b input, in the same technology (estimated from post-layout simulations). These issues, strongly affecting the accuracy and efficiency

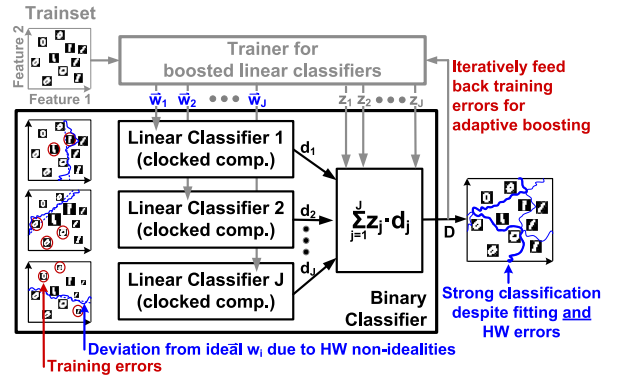


Fig. 6. EACB overcomes circuit non-idealities in weak-classifier implementations.

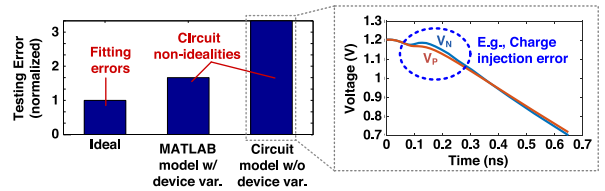


Fig. 7. Comparator-based weak classifiers exhibit significant classification errors in simulation due to numerous circuit non-idealities.

of the classifier, are addressed through the training algorithm discussed next.

IV. TRAINING ALGORITHM

Following from the discussion above, the training algorithm developed for the comparator-based classifier targets three issues: (1) linear classifiers are weak, unable to fit complex data distributions (e.g., insufficient for the image-recognition application demonstration); (2) implementation based on a comparator suffers from numerous analog non-idealities; and (3) the binary-scaled branch NFETs cause exponentially increasing energy/area with the required resolution of weights in \vec{w} . This section describes how these are overcome through training, requiring no additional energy during always-on classification.

A. Error Adaptive Classifier Boosting

The issues of (1) insufficient fitting and (2) analog non-idealities are overcome using EACB [23], which was introduced in Section II-B. As an example, Figure 6 shows a binary classifier formed from the comparator-based structure using EACB (e.g., for 0-vs-1 images). Though the decision boundaries actually applied (shown schematically) deviate from those derived from training, the deviations are compensated by biasing the iterative training of the comparator-based classifiers.

The effect of the deviations is seen in simulations shown in Figure 7, which compares the testing error of an ideal (0-vs-1) linear classifier, a MATLAB model of the comparator structure based on a lookup-table of the NFET branch currents with device variations for a given analog input voltage (extracted from transistor-level Monte Carlo simulations), and a full

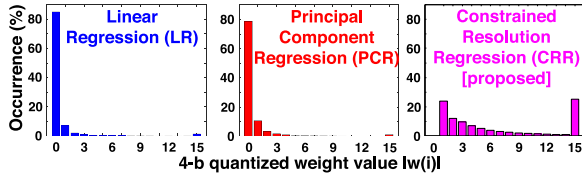


Fig. 8. Histogram of classifier weights (4-b quantization), showing significant negation of many weights in two standard training algorithms, which is resolved in proposed algorithm.

transistor-level circuit model (post-layout) without device variations. As seen, variations in the MATLAB model and non-idealities in the circuit model significantly increase the classification (testing) error.

B. Constrained-Resolution Regression

Regarding the issue of (3) the resolution of \vec{w} 's elements, standard algorithms for training a linear classifier [e.g., Linear Regression (LR) and Principal Component Regression (PCR), which mitigates co-linearity during training] lead to the need for high resolution, for instance >10b in the demonstrated image-recognition application. The reason for this is commonly encountered, and is illustrated in Figure 8. Histograms of \vec{w} 's elements (magnitude only) over all 45 binary classifiers required for the image-recognition application, show that many weights have small values. When quantized (as shown for the example of 4-b resolution), most of these are negated due to a small number of large-valued weights. Thus, error is introduced *after* the learning algorithm, and thus in a way that is highly sub-optimal for fitting the training data.

To substantially reduce the resolution required and thus the overall energy/area of the comparator-based classifier, we propose a training algorithm referred to as CRR. In CRR, such negation is explicitly avoided by constraining the dynamic range of the weights *within* the optimization employed for training the classification model. Thus, the constraint is applied in a way that is optimal for fitting the training data. As shown below, this is achieved by adding an optimization constraint to the linear-regression objective function (where \vec{x}_s is a training feature vector and y_s is the corresponding training label):

$$\begin{aligned} & \text{minimize} \sum_s (y_s - \vec{w} \cdot \vec{x}_s)^2 \\ & \text{subject to } \alpha \leq |w_i| \leq (2^k - 1)\alpha, \quad i = 1, \dots, m. \end{aligned} \quad (2)$$

In the added constraint, k is the resolution desired for binary digital weights, α is a scaling coefficient added to be optimized, and m is the feature-/weight-vector dimensionality. However, the feasible region for the optimization is non-convex due to an absolute value sign in the constraint $|w_i| \geq \alpha$, making the optimization unsolvable via routine quadratic programming. To overcome this, we introduce a binary variable $b_i \in \{0, 1\}$ for the non-convex constraint of each weight w_i , to reformulate $|w_i| \geq \alpha$ to the following two constraints: (1) $w_i + c \cdot b_i \geq \alpha$; (2) $w_i + c \cdot (b_i - 1) \leq -\alpha$. By simply choosing a constant c , which we ensure to have value larger than $\alpha + \max(|w_i|)$, one of these two constraints

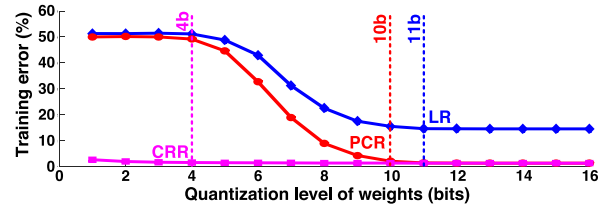


Fig. 9. CRR substantially reduces the resolution required for classifier weights to just 4b.

is instated while the other is trivially satisfied. As a result, for each configuration of \vec{b} (out of 2^m possible configurations), the reformulated optimization problem is convex. The reformulated optimization is:

$$\begin{aligned} & \text{minimize} \sum_s (y_s - \vec{w} \cdot \vec{x}_s)^2 \\ & \text{subject to } -(2^k - 1)\alpha \leq w_i \leq (2^k - 1)\alpha \\ & \quad w_i + c \cdot b_i \geq \alpha \\ & \quad w_i + c \cdot (b_i - 1) \leq -\alpha \\ & \text{where } b_i \in \{0, 1\}, \\ & \quad c > \alpha + \max(|w_i|), \\ & \quad i = 1, \dots, m. \end{aligned} \quad (3)$$

In this form, the optimization is readily solved by mixed-integer-programming solvers [25]. Figure 8 shows the histogram of \vec{w} 's element that results, substantially overcoming the previous problem. Figure 9 shows the classifier performance vs. resolution (for the demonstrated application) using the standard and proposed algorithms. CRR achieves performance at the level of ideal weight precision with just 4-b weights (plus sign).

C. Weight Balancing

A particularly severe non-ideality observed from post-layout simulations of the circuit model is charge-injection error (e.g., shown in the inset of Figure 7). This occurs upon assertion of EN . Pulling down V_S in the comparator-based classifier causes the branch NFETs to transition from sub-threshold to above-threshold, causing charge-injection transients which tends to *increase* the voltage of V_P/V_N (i.e., electrons are pulled from the drain of an NFET into its channel). Due to regeneration, the comparator-based classifier is particularly sensitive to such transients, which thus pose a significant source of decision errors. EACB is limited in overcoming this, because the decision errors bias subsequent iterations to increase the number of NFETs pulling down. While this results in stronger discharge, expected to counter the decision errors, it in fact increases the charge-injection transients, thereby reinforcing the decision errors. Thus we have an instability, which leads to severe imbalance in the number of NFETs driving V_P vs. V_N , and thus imbalance in the charge-injection error.

The effect of imbalanced NFETs on charge injection error is illustrated in Figure 10 based on post-layout simulations. Figure 10a shows a case where the sum of weights for positive

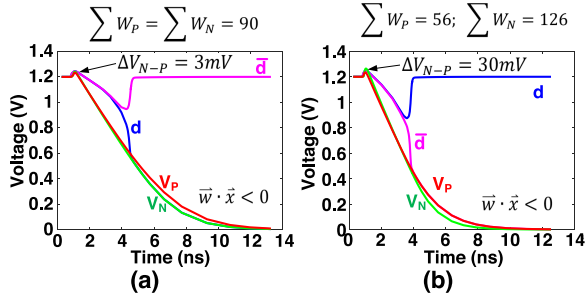


Fig. 10. Comparison of the effect of charge injection errors for (a) balanced NFETs and (b) imbalanced NFETs on V_P and V_N .

and negative branches are equal, thus the number of pull-down NFETs is balanced. The charge injection error is similar on both V_P and V_N nodes, and its influence is largely canceled, yielding the correct decision d for the condition considered ($\vec{w} \cdot \vec{x} < 0$). On the other hand, Figure 10b shows a case where sum of weights for the two branches are very different. In this case, an incorrect decision d is obtained for the condition considered. Indeed, this charge injection is observed to substantially affect classification accuracy in experimental measurements, provided in Section VI-B.

Rather than resorting to circuit-level solutions to address charge injection, which would add additional overhead during actual operation of the classifier, we seek a solution within the training algorithm. In particular, the instability can readily be overcome by adding an additional optimization constraint to counteract the imbalance. The constraint specified below,

$$\sum_{i=1}^m w_i = 0 \quad (4)$$

ensures that the total width of NFET devices on both V_P/V_N is equal, thus giving equal charge-injection error. Since this is a simple linear constraint, it does not significantly increase the complexity of the optimization problem, and simulations show that for an ideal linear classifier (free of charge-injection errors) it does not adversely affect classification accuracy. On the other hand, experimental results from the prototype show that the performance of the comparator-based classifier is substantially improved (Section VI).

V. NOISE MODEL

In this section, a noise model for the comparator-based classifier is developed, and it is later validated through measurements of the prototype. Comparators can be designed to have superior noise efficiency than linear amplifiers [14]. As previously mentioned, the comparator-based classifier can thus replace instrumentation amplifiers, which are generally required in sensing systems. It is worth mentioning that, instrumentation amplifiers sometimes perform other signal processing (e.g., filtering, correlated cancellation of fixed input noise), which would still precede the presented system.

Nonetheless, the efficiency benefit in terms of the system's own noise arises because the output of a linear amplifier is required to settle to a particular value for a given input,

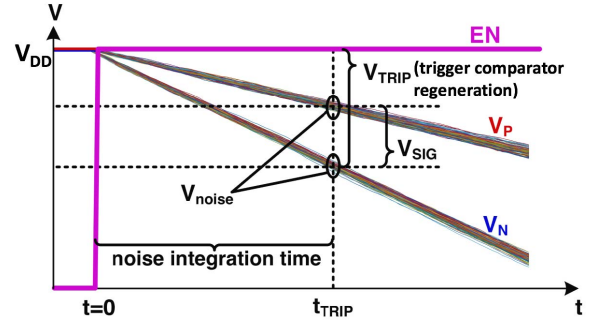


Fig. 11. Illustration of voltage drop variation on summing nodes (V_P/V_N) over time, following the assertion of EN signal. The same process is repeated 1000 times, corresponding to different color V_P/V_N curves.

which is typically achieved through a dominant-pole time constant. For a required settling time, a corresponding noise-bandwidth is incurred. On the other hand, a comparator is not required to settle, but rather only needs to generate a signal larger than a particular threshold in a required amount of time, in order to robustly determine the sign. This implies that an integrator with infinite time constant can be employed (i.e., in Figure 4, I_P/I_N integrate on C_P/C_N to set V_P/V_N). This yields a small noise-bandwidth. It can be shown that for a given transconductance, an integrator gives the shortest time for achieving a desired signal swing [26]. Thus, comparators have the potential for higher speed and lower noise at a given power level.

To develop a noise model for the comparator-based classifier, we employ transient analysis in the time domain, rather than steady-state analysis in the frequency domain, as is typically employed for analysis of linear amplifiers. The reason is that, to first order, for comparators, it is not the noise variance when a steady-state condition is reached that is typically of importance but rather the noise at particular instance in time, namely the point at which regeneration is triggered. For comparators based on integration, the assumption that the noise variance is wide-sense stationary is not necessarily valid, since noise integration causes the variance to change over time. Thus, we start with the time-dependant noise variance, as derived in [14], due to a transconductor g_m whose output current is integrated on a capacitor C_L :

$$V_{noise}^2(t) = \frac{2kT\gamma g_m t}{C_L^2} \quad (5)$$

Corresponding to this, Figure 11 shows the predicted noise variance V_{noise}^2 on the nodes V_P/V_N in the comparator-based classifier. We now define a parameter V_{TRIP} , which represents the drop on V_P/V_N that triggers regeneration. From this, the integration time t_{TRIP} can be defined as follows, where $C_{P,N}$ is the capacitance of the V_P/V_N nodes and $\max\{I_P, I_N\}$ represents the larger of the aggregated pull-down currents from all NFETs on the V_P/V_N nodes:

$$t_{TRIP} = \frac{C_{P,N} V_{TRIP}}{\max\{I_P, I_N\}} \quad (6)$$

Using this value of the integration time, the noise variance on V_P/V_N is determined as follows, where g_m is now the

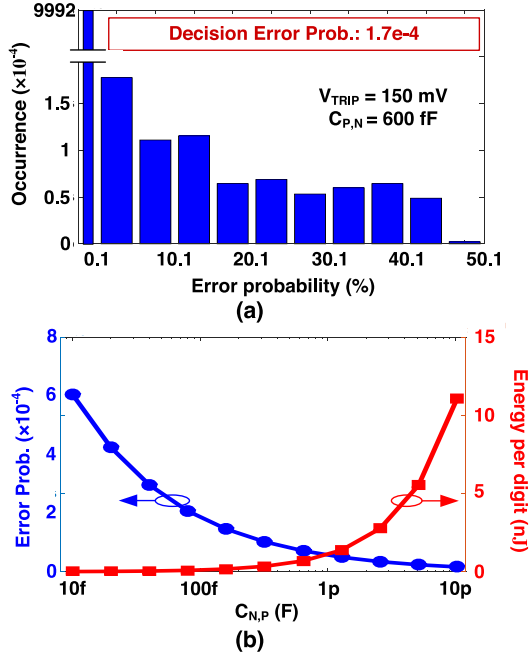


Fig. 12. (a) Low decision-error probability due to noise is achieved for MNIST images; (b) $C_{P,N}$ is a designer parameter for noise-power tradeoff.

aggregated transconductance of all the NFETs pulling down the V_P/V_N nodes:

$$V_{noise}^2 = \frac{2kT\gamma g_m V_{TRIP}}{C_{P,N} \max\{I_P, I_N\}} \quad (7)$$

Note, that this variance models the effect of noise from all of the input NFETs, by representing their aggregated transconductance g_m . Actually, in practice, increasing the NFETs also increases the capacitance of the summing nodes $C_{P,N}$ in proportion, which causes the noise variance to remain unchanged.

Assuming Gaussian noise, the decision-error probability can then be estimated from the above noise variance and the nominal voltage difference between V_P and V_N :

$$V_{SIG} = V_P - V_N|_{t=V_{TRIP}} = V_{TRIP} \left(1 - \frac{\min\{I_P, I_N\}}{\max\{I_P, I_N\}}\right) \quad (8)$$

For a particular application, given the classifier weights derived from training, the values of I_P/I_N can be determined for any set of analog inputs, for instance corresponding to the application dataset. All other parameters (γ , g_m , V_{TRIP} , $C_{P,N}$) in Equation 7 and Equation 8 can be determined from the device or circuit parameters. As an example, Figure 12a shows a histogram of error probability for the 45 binary classifiers required in the image-recognition application demonstrated, over 10,000 images in the MNIST dataset. With $V_{TRIP} \approx 150\text{mV}$ (from simulation) and $C_{P,N} \approx 600\text{fF}$ (from post-layout extraction), the decision-error probability due to noise is low, with average value $1.7\text{e-}4$. We point out that $C_{P,N}$ presents a designer parameter (as can V_{TRIP} , through circuit modification). This gives a tradeoff between error probability due to noise and energy consumption, as shown in Figure 12b for the MNIST dataset. Additionally, a larger $C_{P,N}$ also implies a larger area, posing an additional design tradeoff.

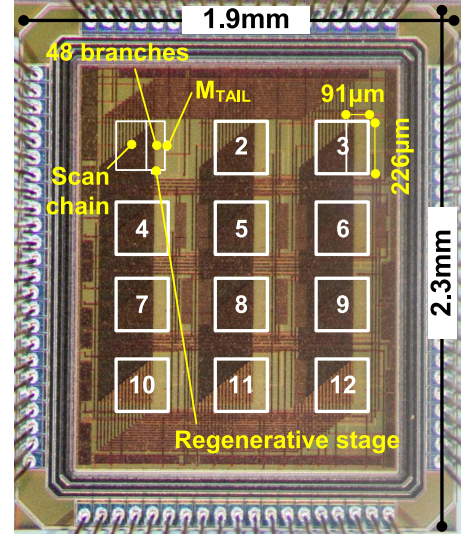


Fig. 13. Die photograph of IC, implemented in 130nm CMOS.

TABLE I
MEASUREMENT SUMMARY OF IC

Technology / Supply voltage	130nm/1.2V
Speed (1.2V)	1.3MHz
Area per classifier	$91 \times 226\mu\text{m}^2$
Sensor input range	0.3 – 1.2V
Total noise σ due to all inputs	$280\mu\text{V}_{RMS}$
Detection accuracy (MNIST)	90%
Total # binary classifiers	45
Branches per classifier (m)	48
Average # weak classifiers	4.4
Energy per comparator (1.2V)	2.43pJ
Energy per digit classification	534pJ
Total energy savings (1.2V)	$33\times$

VI. PROTOTYPE MEASUREMENTS

Figure 13 shows the die photo and Table I shows measurement summary of the prototype IC. The IC is implemented in 130nm CMOS and integrates 12 comparator-based classifiers, along with a shift register for loading 5-b weights (4-b magnitude plus 1-b sign) from training. As previously mentioned, each comparator consists of $m = 48$ NFET branches, supporting 48 analog-input feature channels. The area of each 48-branch comparator-based classifier is $91 \times 226\mu\text{m}^2$. Comparison can be performed at 1.3M classifications per second ($V_{DD} = 1.2\text{V}$) at an energy of 2.43pJ per comparator-based classifier. In the following subsections, measurements are presented to validate the noise model and an application demonstration. Measurement results are validated across 10 chips, with multiples runs each. Similar results are obtained in all cases.

A. Noise Model Validation

The noise model presented in Section V is validated experimentally. Figure 14a shows the experimental setup. A comparator-based classifier is configured to have

24 activated positive and negative branches, each with weight-magnitude value of 7 (4'b0111), i.e., mid-point of weight-magnitude range. The analog inputs corresponding to the positive and negative branches are driven by a slow staircase ramp using a 16-b DAC (with step size and noise verified to be below the noise of the comparator-based classifier). Then, 10k comparisons are performed at each stair-case step to derive the probability of the output decision d being 1. This yields the measured grey curves shown in Figure 14b, which are taken for different input common-mode levels; since changing the common-mode level changes the g_m and I_D of the NFET branches, which are parameters affecting the noise, this enables validation of the noise model developed. In particular, the measured curves are seen to exhibit different shapes, corresponding to the offset and variance due to noise of the comparator-based classifier. For validation of the noise model, the curves predicted from the noise model (Equation 13) are shown in blue, and the measured curves after removing offsets are shown in red. Figure 14c explicitly shows the the variances predicted by the noise model and extracted from measurements for various input common-mode levels. As seen, good agreement between the model and measurements is observed.

The blue curves for predicted noise model in Figure 14b are derived as following. As shown in Figure 14a, define mean analog inputs corresponding to positive and negative branches as $x_{in,P/N}$ (here we use shorthand notation $x_{in,P/N}$ to represent $x_{in,P}$ and $x_{in,N}$, which also applies below). The noise on the integration nodes $V_{P/N}$ can be input-referred, leading to input-plus-noise signals $x_{P/N}$. Approximating the gain A from the inputs to the integration nodes to be

$$A = \frac{V_{P/N}}{x_{P/N}} = -\frac{g_m t_{TRIP}}{C_{P,N}}, \quad (9)$$

the input referred noise can be derived from Equation 7 as

$$V_{noise,in}^2 = \frac{V_{noise}^2}{A^2} = \frac{2kT\gamma \max\{I_P, I_N\}}{g_m C_{P,N} V_{TRIP}}. \quad (10)$$

For simplicity, we assume the input-referred noise to be Gaussian:

$$x_{P/N} \sim N(x_{in,P/N}, V_{noise,in}^2). \quad (11)$$

As a result, the differential input-plus-noise signal x_{Diff} is a random variable with the following distribution:

$$x_{Diff} = x_P - x_N \sim N(x_{in,P} - x_{in,N}, 2V_{noise,in}^2). \quad (12)$$

The probability that output d is 1 corresponds to the probability that x_{Diff} is positive:

$$P(d=1) = P(x_{Diff} > 0) = \Phi\left(\frac{x_{in,P} - x_{in,N}}{\sqrt{2}V_{noise,in}}\right), \quad (13)$$

which can be numerically computed given the model parameters).

B. Image-Recognition Demonstration

For demonstration, image-recognition is performed to detect 0-9 numerical handwritten digits from the MNIST dataset [7]. Though handwriting recognition is not a typical

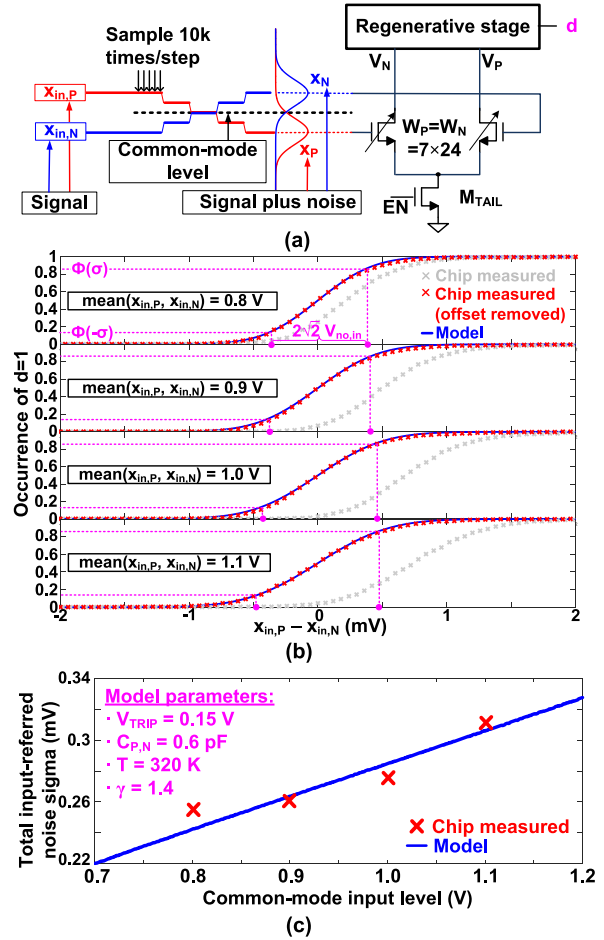


Fig. 14. Validation of the noise model. (a) experimental setup; (b) computation of the input referred noise model; (c) experimental results comparing chip measured result versus model derived computation.

IoT application, it provides a benchmark for low-power image detection, which is generally regarded as an important IoT capability [27]. The proposed system is envisioned to directly interface with analog imagers (possibly on the same die or within the same package), where each pixel output can directly drive one of the comparator's NFET branches.

Figure 15 shows the system implementation. For 10-way classification all-versus-all (AVA) voting is performed over 45 binary classifiers corresponding to all pairs of digits. Each binary classifier is implemented as boosted comparator-based classifiers, trained using the algorithms presented in Section IV. The frequency with which EACB training must be performed depends on how stationary the hardware non-idealities are [23]. In the demonstrated system, training is performed once (hardware non-idealities are static between training and testing). On the other hand, the classification system (shown in solid black in Figure 15) runs continuously in real time.

Figure 16 shows details of the experimental setup. While MNIST images correspond to 28×28 pixels, the prototype IC supports up to $m = 48$ analog sensor-input features (due to pin limitations). So images are resized from 28×28 to 9×9 pixels by low-pass filtering, and then further down selecting

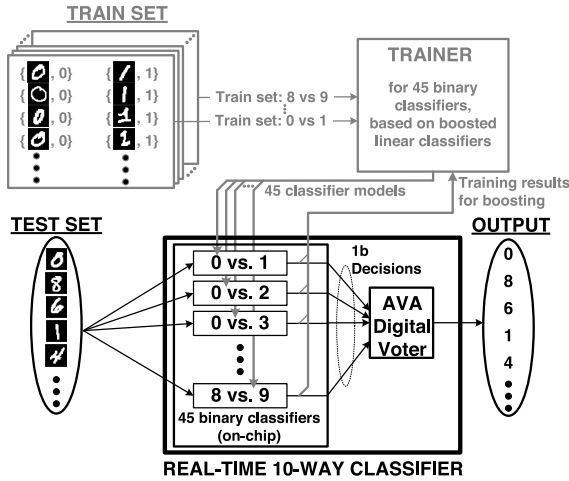


Fig. 15. Demonstrated system performs 0-9 digit recognition by taking analog pixel data from images in the MNIST dataset. 10-way classification is based on all-vs-all voting over 45 binary classifiers. Training (shown in grey) is performed once offline (as described in Section IV).

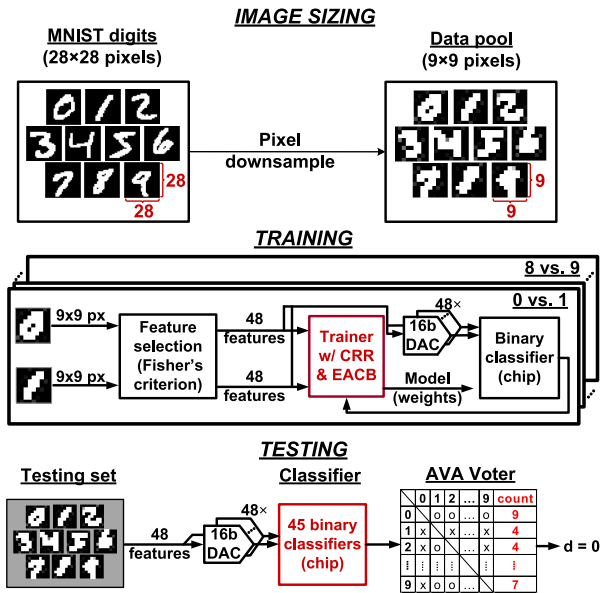


Fig. 16. Details of the experimental setup for hand written digits recognition.

from 81 to 48 features, using the widely employed Fisher's criterion for feature selection [28]. Reducing the number of image-pixel features in this way takes the classification performance of an ideal (ADC/digital-MAC) system based on boosted linear classifiers from $\sim 96\%$ to 90% , making this the target for the comparator-based system. In an eventual system, pin limitations could be addressed by integrating the image sensor on the same die or in the same package as the classification system. To evaluate classification performance, 5-fold cross-validation of training and testing is performed, by feeding the pixel features to the prototype IC via 16-b DACs. Additions required for boosting and all-versus-all voting are performed off chip from the 1-b comparator-based-classifier decisions measured (energy of additions is considered below, as described).

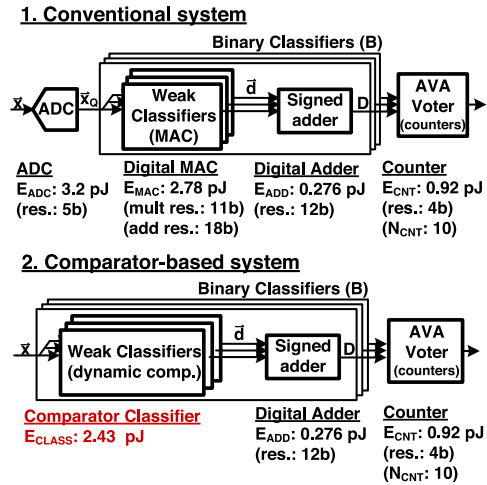


Fig. 17. Assumed energy numbers for a conventional system (consisting of ADC and digital MACs) and the demonstrated system.

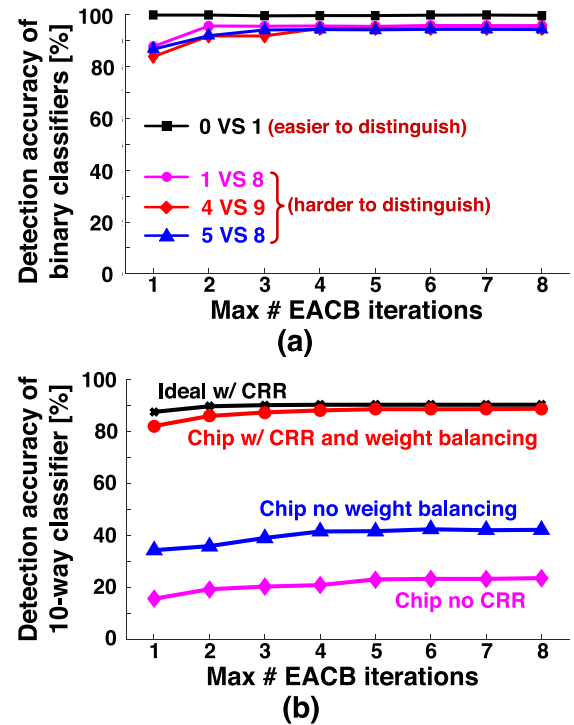


Fig. 18. Measured performance (versus EACB iterations) of comparator-based classifier for (a) a few example binary classifiers, and (b) 10-way classifier formed from AVA voting of 45 binary classifiers.

Figure 17 shows details of the methodology used for energy comparison of the demonstrated system versus a conventional ADC/digital-MAC system. The component bit precisions required are determined by performing simulations in MATLAB. For the conventional system, the bit-precision for the ADCs and MACs are optimized to the critical bit-precision required without significantly degrading classification performance, thus corresponding to the lowest energy digital implementation to the best of our effort. All component energies assume a supply voltage of 1.2V and implementation in a

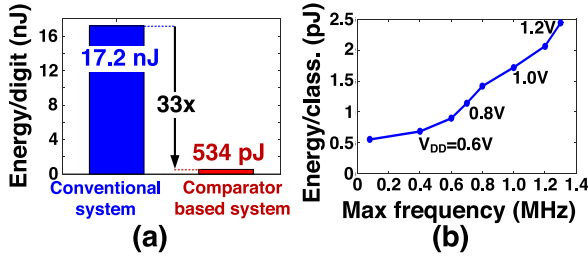


Fig. 19. (a) Total energy savings of the proposed system compared to a corresponding optimized digital implementation; (b) energy-versus-frequency scaling.

TABLE II
SUMMARY OF PERFORMANCE

Metrics	[29]	[30]	This Work
Technology	65nm	40nm	130nm
Implementation	$\Delta\Sigma$ Modulator	Switched Caps	Comparator
Multiplier Resolution	14b	3b	5b
Accuracy (MNIST)	88%	—	90%
E_{MAC} (pJ)	2.44	0.11	0.051
Speed (MACs/sec)	100M	1G	63M
Area (mm^2)	0.0594	0.012	0.0206

* In terms of integration level, [29] performs combined PCA and linear SVM classification on-chip, while ECOC-based 10-class classification is performed off-chip. [30] employs a 3-layer convolutional neural network, with the proposed switched capacitor matrix multiplier only applied to the first convolution layer. The subsequent stages (one activation layer, one convolution and activation layer, one fully connected layer) are all implemented in digital off-chip. This work performs all binary classifications on-chip. While classifier ensemble (12b adder) and all-versus-all voting (4b counter) are performed off-chip.

130nm CMOS technology, with values taken from measured chips (previously reported) or from post-layout simulations performed [1], [6]. Note, instrumentation amplifiers are not considered in the conventional system. Image sensor inputs are assumed to be in the range of 0.3V-1.2V. In a system requiring the instrumentation amplifiers, even greater energy savings would be expected from the comparator-based system due to better noise efficiency (Section V).

Figure 18a shows binary classification performance for a few digit pairs. Significant boosting is observed. Figure 18b shows the measured 10-way digit-recognition performance, versus number of EACB iterations. While the conventional system with CRR training (assuming ideal implementation), achieves performance convergence (to 90%) in an average of 2.8 iterations across the 45 binary classifiers, the comparator-based system requires an average of 4.4 iterations. This highlights the importance of EACB for overcoming implementation non-idealities. Further, the performance of the comparator-based system without CRR and without weight balancing are also shown respectively, highlighting the importance of CRR and weight balancing.

At a nominal V_{DD} of 1.2V, with the energy per decision of 2.43pJ for a comparator-based classifier, the total energy for 10-way digit classification with average of 4.4 EACB iterations (and including the additions required for boosting and AVA voting) is 534pJ. As shown in Figure 19a, this corresponds to 33 \times lower energy than the conventional system with average of 2.8 EACB iterations. Finally, Figure 19b shows how the

energy per decision for the comparator-based classifiers can be reduced at the cost of lower speed by scaling V_{DD} .

Table II provides a performance comparison with other works, focusing on analog classification. In addition to low energy per MAC, the presented design achieves high classification accuracy even at low resolutions, thanks to the proposed training algorithm.

VII. CONCLUSION

This paper presented a classifier that replaces the need for low-noise instrumentation amplifiers, ADCs, and digital MACs, replacing these with clocked comparators that consume only CV^2 energy. Analog computation introduced non-idealities and severe energy/area scaling with the required bit resolution of the classification model. Both of these challenges were overcome via the classifier training algorithm. Analog non-idealities were overcome using a previous boosting algorithm we presented called Error-Adaptive Classifier Boosting. Resolution requirements of classification model were overcome with a new weak-classifier training algorithm presented, called Constrained-Resolution Regression. The prototype demonstrated 10-way image classification of numerical digits (from images of the MNIST dataset down-sampled/downselected to 48 pixels), achieving performance at the level of an ideal ADC/digital-MAC implementation (i.e., detection accuracy of 90%), yet at 33 \times lower energy.

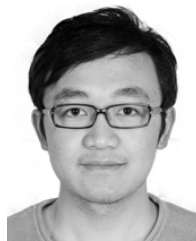
ACKNOWLEDGMENT

The authors thank MOSIS for IC fabrication.

REFERENCES

- [1] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.
- [2] Y.-H. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 262–263.
- [3] S. Park, S. Choi, J. Lee, M. Kim, J. Park, and H.-J. Yoo, "A 126.1 mW real-time natural UI/UX processor with embedded deep-learning core for low-power smart glasses," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2016, pp. 254–255.
- [4] S. Hanson and D. Sylvester, "A 0.45–0.7 V sub-microwatt CMOS image sensor for ultra-low power applications," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2009, pp. 176–177.
- [5] N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Gutttag, and A. P. Chandrakasan, "A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, Apr. 2010.
- [6] Z. Wang, J. Zhang, and N. Verma, "Realizing low-energy classification systems by implementing matrix multiplication directly within an ADC," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 6, pp. 825–837, Dec. 2015.
- [7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [8] K. Kang and T. Shibata, "An on-chip-trainable Gaussian-kernel analog support vector machine," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1513–1524, Jul. 2010.
- [9] J. Lu, S. Young, I. Arel, and J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 270–281, Jan. 2015.

- [10] J. Zhang, Z. Wang, and N. Verma, "A machine-learning classifier implemented in a standard 6T SRAM array," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2016, pp. 1–2.
- [11] W. Rieutort-Louis, T. Moy, Z. Wang, S. Wagner, J. C. Sturm, and N. Verma, "A large-area image sensing and detection system based on embedded thin-film classifiers," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 281–290, Jan. 2016.
- [12] M. A. B. Altaf and J. Yoo, "A 1.83 μ J/classification, 8-channel, patient-specific epileptic seizure classification SoC using a non-linear support vector machine," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 49–60, Feb. 2016.
- [13] E. Yao and A. Basu, "VLSI extreme learning machine: A design space exploration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 1, pp. 60–74, Jan. 2017.
- [14] T. Sepke, P. Holloway, C. G. Sodini, and H.-S. Lee, "Noise analysis for comparator-based circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 541–553, Mar. 2009.
- [15] R. Shapeshkar, *Ultra Low Power Bioelectronics: Fundamentals, Biomedical Applications, and Bio-inspired Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
- [16] Z. Wang, J. Zhang, and N. Verma, "Reducing quantization error in low-energy FIR filter accelerators," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 1032–1036.
- [17] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 10, pp. 930–936, Oct. 2001.
- [18] V. K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S. T. Chakradhar, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency," in *Proc. 47th ACM/IEEE Design Autom. Conf.*, Jun. 2010, pp. 555–560.
- [19] Y. Yetim, M. Martonosi, and S. Malik, "Extracting useful computation from error-prone processors for streaming applications," in *Proc. Design, Autom. Test Eur. Conf.*, 2013, pp. 202–207.
- [20] L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2010, pp. 1560–1565.
- [21] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.
- [22] Z. Wang, K. H. Lee, and N. Verma, "Overcoming computational errors in sensing platforms through embedded machine-learning kernels," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1459–1470, Aug. 2015.
- [23] Z. Wang, R. E. Schapire, and N. Verma, "Error adaptive classifier boosting (EACB): Leveraging data-driven training towards hardware resilience for signal inference," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 4, pp. 1136–1145, Apr. 2015.
- [24] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2012.
- [25] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Inc., Houston, TX, USA, 2012.
- [26] J.-T. Wu and B. A. Wooley, "A 100-MHz pipelined CMOS comparator," *IEEE J. Solid-State Circuits*, vol. 23, no. 6, pp. 1379–1385, Dec. 1988.
- [27] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.
- [28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
- [29] F. N. Buhler, A. E. Mendrela, Y. Lim, J. A. Fredenburg, and M. P. Flynn, "A 16-channel noise-shaping machine learning analog-digital interface," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2016, pp. 1–2.
- [30] E. H. Lee and S. S. Wong, "A 2.5 GHz 7.7 TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40 nm," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 418–419.



Zhuo Wang received the B.S. degree in micro-electronics from Peking University, Beijing, China, in 2011, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, in 2013 and 2017, respectively. He is currently a Research Staff Member with the IBM Watson Research Center. His research focuses on leveraging statistical approaches, such as machine learning, for achieving hardware relaxation in an algorithmic and architectural level in resource-constrained platforms, such as embedded sensing systems. He was the recipient of the 2011 Peking University Best Undergraduate Dissertation Award, the 2014 ICASSP Conference Best Paper Award Nomination, and the 2015 Princeton University Honorific Fellowship.



Naveen Verma received the B.A.Sc. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, in 2005 and 2009, respectively. Since 2009, he has been with the Department of Electrical Engineering, Princeton University, where he is currently an Associate Professor. His research focuses on advanced sensing systems, including low-voltage digital logic and SRAMs, low-noise analog instrumentation and data-conversion, large-area sensing systems based on flexible electronics, and low-energy algorithms for embedded inference, especially for medical applications.

Prof. Verma is a Distinguished Lecturer of the IEEE Solid-State Circuits Society, and serves on the technical program committees for the International Solid-State Circuits Conference, the VLSI Symposium, DATE, and the IEEE Signal-Processing Society (DISPS). He was a recipient or co-recipient of the 2006 DAC/ISSCC Student Design Contest Award, the 2008 ISSCC Jack Kilby Paper Award, the 2012 Alfred Rheinlein Junior Faculty Award, the 2013 NSF CAREER Award, the 2013 Intel Early Career Award, the 2013 Walter C. Johnson Prize for Teaching Excellence, the 2013 VLSI Symposium Best Student Paper Award, the 2014 AFOSR Young Investigator Award, the 2015 Princeton Engineering Council Excellence in Teaching Award, and the 2015 IEEE Trans. CPMT Best Paper Award.