

# Analog Gated Recurrent Unit Neural Network for Detecting Chewing Events

Kofi Odame, *Senior Member, IEEE*, Maria Nyamukuru, *Student Member, IEEE*, Mohsen Shahghasemi, *Student Member, IEEE*, Shengjie Bi, David Kotz, *Fellow, IEEE*

**Abstract**—We present a novel gated recurrent neural network to detect when a person is chewing on food. We implemented the neural network as a custom analog integrated circuit in a  $0.18\ \mu\text{m}$  CMOS technology. The neural network was trained on 6.4 hours of data collected from a contact microphone that was mounted on volunteers' mastoid bones. When tested on 1.6 hours of previously-unseen data, the analog neural network identified chewing events at a 24-second time resolution. It achieved a recall of 91% and an F1-score of 94% while consuming  $1.1\ \mu\text{W}$  of power. A system for detecting whole eating episodes—like meals and snacks—that is based on the novel analog neural network consumes an estimated  $18.8\ \mu\text{W}$  of power.

**Index Terms**—Eating detection, wearable devices, analog LSTM, neural networks.

## I. INTRODUCTION

Monitoring food intake and eating habits are important for managing and understanding obesity, diabetes and eating disorders [1], [2], [3]. Because self-reporting is unreliable, many wearable devices have been proposed to automatically monitor and record individuals' dietary habits [4], [5], [6], [7]. The challenge is that these devices store or transmit raw data for offline processing. This is a power-consumptive approach that requires a bulky battery or frequent charging, which intrudes on the user's normal daily activities and is thus prone to poor user adherence and acceptance [8], [9], [10], [11].

We recently addressed this problem with a long short-term memory (LSTM) neural network for eating detection that can be embedded on the wearable device [12], [13]. However, that approach required a power-consumptive analog-to-digital converter (ADC). It also required the microcontroller unit (MCU) to unnecessarily spend power processing irrelevant data.

Analog LSTM neural networks have been proposed as a way to eliminate the ADC and also to minimize the microcontroller's processing of irrelevant data. Unfortunately, the state-of-the-art analog LSTMs [14], [15], [16], [17], [18] are implemented with operational amplifiers (opamps), current/voltage converters, Hadamard multiplications and internal ADCs and digital-to-analog converters (DACs). These peripheral components represent a significant amount of overhead cost in terms

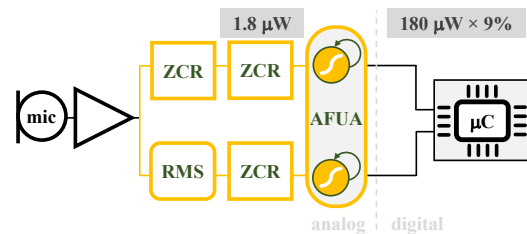


Fig. 1. Block diagram of proposed eating detection system. From the contact microphone output, the ZCR and RMS blocks extract features based on zero-crossing rate and root-mean-square. The analog neural network (labelled 'AFUA') processes these features and produces a one-hot encoded output that predicts the presence or absence of a chewing event. The microcontroller (' $\mu\text{C}$ ') merges and filters the individual chewing events into whole eating episodes. The analog signal processing chain up to the AFUA block consumes  $1.8\ \mu\text{W}$  of power. The microcontroller is active only 9% of the time, during which it consumes  $180\ \mu\text{W}$  of power.

of power consumption, which diminishes the benefits of an analog LSTM (see Table I).

In this paper, we present the design, implementation, analysis and measurement results of a novel analog integrated circuit LSTM for embedded eating event detection that eliminates the need for a power-consumptive ADC. Unlike previous analog LSTM implementations, our solution contains no internal DACs, ADCs, opamps or Hadamard multiplications. Our novel approach is based on a current-mode adaptive filter, and it eliminates over 90% of the power requirements of a more conventional solution.

## II. EATING DETECTION SYSTEM

Figure 1 shows our proposed Adaptive Filter Unit for Analog (AFUA) long short-term memory as part of a signal processing system for detecting eating episodes. The input to the system is produced by a contact microphone that is mounted on the user's mastoid bone. Features are extracted from the contact microphone signal and input to the AFUA neural network, which infers whether or not the user is chewing. The AFUA's output is a one-hot encoding  $((2, 0)=\text{chewing}; (0, 2)=\text{not chewing})$  of the predicted class label. Finally, a microcontroller processes the predicted class labels and groups the chewing events into discrete eating episodes, like a meal, or a snack [4], [5]. Following is a detailed description of the feature extraction and neural network components of the system.

Manuscript received X; revised Y; accepted Z

Kofi Odame, Maria Nyamukuru and David Kotz are with Dartmouth College Hanover, New Hampshire, United States of America (email: odame@dartmouth.edu). Mohsen Shaghsemi and Shengjie Bi, formerly of Dartmouth College, are now with Apple Inc. and Meta Platforms Inc., respectively.

Architecture	LSTM Type	$m \times n$	Power Consumption Overhead (%)				
			ADC	DAC	Buffer	Opamp, V/I	Total
This work	AFUA	$10 \times 16$	0	0	3	0	3
[19]	GRU	$10 \times 16$	0	0	32	0	32
[17]	Classical LSTM	$128 \times 128$	12	25	1	30	68
[18]	Classical LSTM	$16 \times 16$	3	17	8	1	29

TABLE I

THE PROPOSED LSTM ('AFUA') HAS THE FEWEST PERIPHERAL COMPONENTS AND HENCE THE LOWEST POWER CONSUMPTION OVERHEAD (SEE SECTION IV-A DERIVATION).  $m$  AND  $n$  ARE NUMBER OF HIDDEN STATES AND INPUTS, RESPECTIVELY. NOTE: FOR A GIVEN ANALOG LSTM ARCHITECTURE, THE LARGER THE  $m \times n$  PRODUCT, THE SMALLER THE OVERHEAD. FOR FAIR COMPARISON, WE REPORT AFUA OVERHEAD COST FOR  $m \times n = 10 \times 16$ . THE CORE PROCESSING BLOCK FOR ALL THE LSTM ARCHITECTURES IS BASED ON THE SAME BASIC VECTOR MATRIX MULTIPLIER (VMM) STRUCTURE. IF WE IMPLEMENT EACH ARCHITECTURE IN THE SAME PROCESS TECHNOLOGY NODE WITH THE SAME VMM, THEN THE ARCHITECTURE WITH THE LEAST OVERHEAD WILL CONSUME THE LEAST TOTAL POWER.

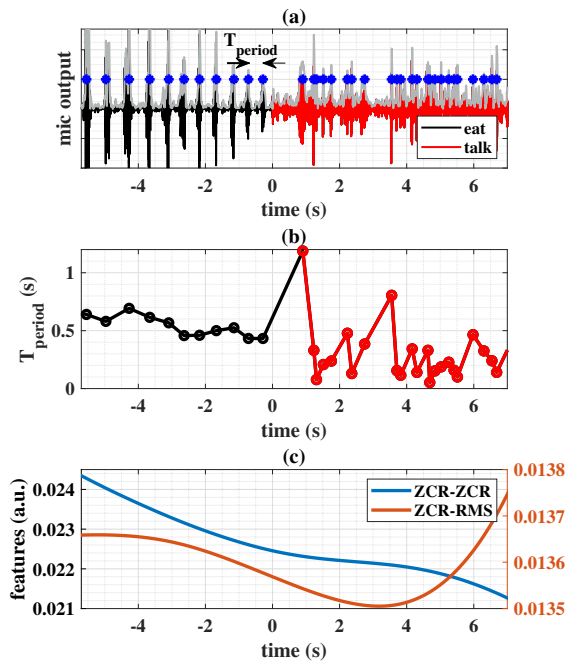


Fig. 2. Typical time series data for chewing and talking events. (a) Data from contact microphone shows that chewing (time < 0 s) is characterized by quasi-periodic bursts. No quasi-periodicity is observed during talking (time  $\geq 0$  s). (b) Duration between signal bursts (' $T_{\text{period}}$ '). For the chewing event (time < 0 s),  $T_{\text{period}}$  is relatively constant. In contrast,  $T_{\text{period}}$  varies widely during the talking event. (c) Features extracted from microphone output.

### A. Feature Extraction

As demonstrated in Fig. 2, chewing is characterized by quasi-periodic bursts of large amplitude, low frequency signals that can be measured by a contact microphone or accelerometer that is mounted on the head [12], [5]. We can use the root mean square (RMS) and the zero-crossing rate (ZCR) to capture the signal's amplitude and frequency, respectively. A second ZCR operation applied to the RMS and the initial ZCR will produce information about the signal's periodicity. The RMS block is simply an envelope detector [20]. The ZCR block comprises a zero-crossing detector [21] followed by a bandlimited transconductance amplifier that integrates the detected zero crossings over time.

### B. Analog LSTM

Fundamentally, an LSTM is a neuron that selectively retains, updates or erases its memory of input data [22]. The gated recurrent unit (GRU) is a simplified version of the classical LSTM, and it is described with the following set of equations [23]:

$$r_j = \sigma([\mathbf{W}_r \mathbf{x}]_j + [\mathbf{U}_r \mathbf{h}_{(t-1)}]_j) \quad (1)$$

$$z_j = \sigma([\mathbf{W}_z \mathbf{x}]_j + [\mathbf{U}_z \mathbf{h}_{(t-1)}]_j) \quad (2)$$

$$\tilde{h}_j^{(t)} = \tanh([\mathbf{W} \mathbf{x}]_j + [\mathbf{U}(\mathbf{r} \odot \mathbf{h}_{(t-1)})]_j) \quad (3)$$

$$h_j^{(t)} = z_j h_j^{(t-1)} + (1 - z_j) \tilde{h}_j^{(t)}, \quad (4)$$

where  $\mathbf{x}$  is the input,  $h_j$  is the hidden state,  $\tilde{h}_j$  is the candidate state,  $r_j$  is the reset gate and  $z_j$  is the update gate. Also,  $\mathbf{W}_*$  and  $\mathbf{U}_*$  are learnable weight matrices.

To implement the GRU in an efficient analog integrated circuit that contains no DACs, ADCs, operational amplifiers or multipliers, we can transform Eqn. (1)-(4) as follows. The  $\sigma$  function of Eqn. (2) gives  $z_j$  a range of (0, 1), and the extrema of this range reveals the basic mechanism of the update equation, Eqn. (4). For  $z_j = 0$ , the update equation is  $h_j^{(t)} = \tilde{h}_j^{(t)}$ . For  $z_j = 1$ , the update equation becomes  $h_j^{(t)} = h_j^{(t-1)}$ . Without loss of generality, we can replace  $(1 - z_j)$  with  $z_j$  (this merely inverts the logic of the update gate, and inverts the sign of the  $\mathbf{W}_z$  and  $\mathbf{U}_z$  weight matrices). So, replacing  $(1 - z_j)$  and rearranging the update equation gives us

$$(h_j^{(t)} - h_j^{(t-1)}) / z_j + h_j^{(t-1)} = \tilde{h}_j^{(t)}, \quad (5)$$

which is simply a first-order low pass filter with a continuous-time form of

$$\frac{\tau}{z_j(t)} \frac{dh_j}{dt} + h_j(t) = \tilde{h}_j(t), \quad (6)$$

where  $\tau = \Delta T$ , the time step of the discrete-time system. The gating mechanics of the continuous- versus discrete-time update equations are equivalent, modulo the inverted logic: For  $z_j(t) = 0$ , Eqn. (6) is a low-pass filter with an infinitely large time constant, and  $h_j(t)$  does not change (this is equivalent to  $h_j^{(t)} = h_j^{(t-1)}$  in discrete time). For  $z_j(t) = 1$ , Eqn. (6) is a low-pass filter with a time constant of  $\tau = \Delta T$ . Since the  $\Delta T$  time step is small relative to the GRU's dynamics, a time constant of  $\tau = \Delta T$  produces  $h_j(t) \approx \tilde{h}_j(t)$  (equivalent to  $h_j^{(t)} = \tilde{h}_j^{(t)}$  in discrete time).

Various studies [24], [25], [26], [13] have found the reset gate unnecessary with slow-changing signals, and for event detection. As these scenarios describe our eating detection application, we can discard the reset gate.

Finally, if we translate the origins [27] of both  $h_j(t)$  and  $\tilde{h}_j(t)$ , then we can replace the  $\tanh$  with a saturating function that has a range of  $(0, 1)$ . Such a saturating function can easily be implemented in analog circuitry, by taking advantage of the unidirectional nature of a transistor's drain-source current. We replace both the  $\tanh$  and the  $\sigma$  with the following saturating function,

$$f(y) = \frac{\max(y, 0)^2}{1 + \max(y, 0)^2}, \quad (7)$$

translate the origin and discard the reset gate to arrive at the *Adaptive Filter Unit for Analog LSTM* (AFUA):

$$z_j = f([\mathbf{W}_z \mathbf{x}]_j + [\mathbf{U}_z (\mathbf{h} - \mathbf{1}) + \mathbf{b}_z]_j) \quad (8)$$

$$\tilde{h}_j = f([\mathbf{W} \mathbf{x}]_j + [\mathbf{U} (\mathbf{h} - \mathbf{1}) + \mathbf{b}]_j) \quad (9)$$

$$\frac{\tau}{z_j} \frac{dh_j}{dt} = 2\tilde{h}_j - h_j, \quad (10)$$

where  $[\cdot]_j$  is the  $j$ 'th element of the vector. Also,  $\mathbf{x}$  is the input,  $h_j$  is the hidden state and  $\tilde{h}_j$  is the candidate state. The variable  $\tau$  is the nominal time constant, while  $z_j$  controls the state update rate in Eqn. (10).  $\mathbf{W}_z$ ,  $\mathbf{U}_z$ ,  $\mathbf{W}$ ,  $\mathbf{U}$  are learnable weight matrices, while  $\mathbf{b}_z$ ,  $\mathbf{b}$  are learnable bias vectors. Simulation results (Fig. 3) for a multi-class machine learning task show that the AFUA performs with a comparable level of accuracy as the GRU and classical LSTM.

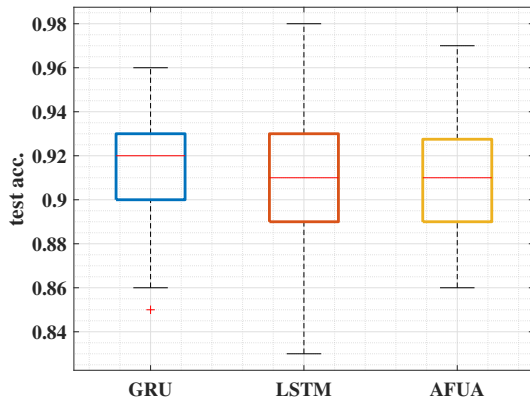


Fig. 3. Simulation results (test set accuracy) for 10-class keyword spotting task [28], [29]. The simulated neural network architecture comprises: a 16-unit LSTM input layer, a second 16-unit LSTM layer, a 10-unit dense layer (ReLU activations) and a 10-unit dense output layer (softmax activations). We implemented the LSTM layers first with GRU [23], then classical LSTM [22] and finally AFUA neurons.

### III. ANALOG LSTM CIRCUIT IMPLEMENTATION

Figure 4 shows the high-level block diagram of the AFUA neural network. It comprises two AFUA cells (with corresponding hidden states  $h_0$  and  $h_1$ ), and it accepts two inputs,  $x_0$  and  $x_1$ . Unlike previous LSTMs [14], [15], [16], [17], [18], the AFUA network contains no digital-to-analog converters,

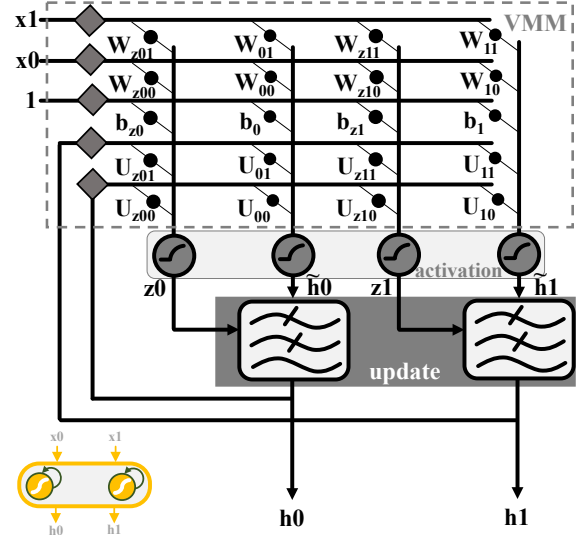


Fig. 4. High level architecture of the AFUA neural network, which has a two-dimensional input feature vector,  $\mathbf{x} = [x_0, x_1]^T$ . The network keeps a memory of past inputs by feeding back its hidden states,  $h_0, h_1$ , to the vector matrix multiplier (VMM). The persistence of the network's memory depends on the time constants,  $z_0, z_1$ , of the adaptive low pass filters in the 'update' block. Finally, the 'activation' block provides saturating nonlinearities described by Eqn. (7).

analog-to-digital converters, operational amplifiers or four-quadrant multipliers. Avoiding these power-consuming components is what makes the AFUA implementation so efficient. Following are the circuit implementation details of the AFUA.

#### A. Dimensionalization

To realize the AFUA Eqns. (8), (9), (10) and (7) as an analog circuit, we first 'dimensionalize' each variable and implement it as the ratio of a time-varying current and a fixed unit current,  $I_{\text{unit}}$  [30], [31]. For instance, we represent the update gate variable,  $z_j$ , as  $I_z/I_{\text{unit}}$ .

#### B. Activation Function

The Eqn. (7) function is implemented as the current-starved current mirror shown in Fig. 5. Kirchhoff's Current Law applied to the source of transistor  $M_3$  gives

$$I_{\text{out}} = I_3 = I_{\text{unit}} - I_4. \quad (11)$$

The transistors are all sized equally, meaning that, from Kirchhoff's Voltage Law, the gate source voltage of transistor  $M_3$  is

$$V_{GS3} = 2V_{GS1} + V_{GS4} - 2V_{GSa}, \quad (12)$$

where we have assumed that the body effect in  $M_2$  and  $M_b$  is negligible. If we operate the transistors in the subthreshold region, then Eqn. (12) implies

$$I_{\text{out}} = I_3 = \frac{I_4 I_1^2}{I_{\text{unit}}^2}. \quad (13)$$

Combining Eqns. (11) and (13) gives us

$$I_{\text{out}} = \frac{I_{\text{unit}} I_1^2}{I_{\text{unit}}^2 + I_1^2}. \quad (14)$$

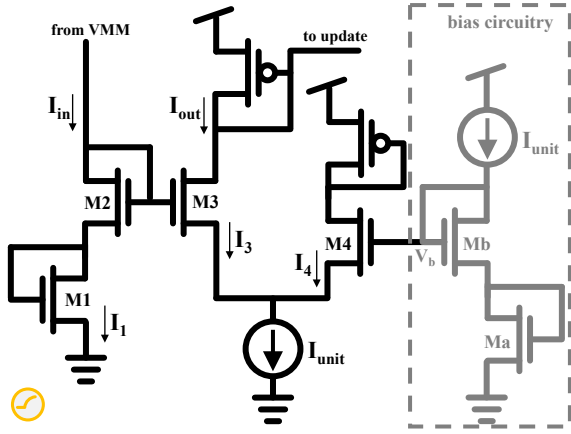


Fig. 5. Activation function circuit schematic. A version of the input signal,  $I_{in}$ , is reflected as current  $I_{out}$ . The tail bias current source of the M3-M4 differential pair limits the output current to  $I_{out} < I_{unit}$ . Also, the one-sidedness of the nMOS drain current limits  $I_{out}$  to positive values only. In summary, the activation function circuit produces  $0 \text{ A} \leq I_{out} < I_{unit}$ .

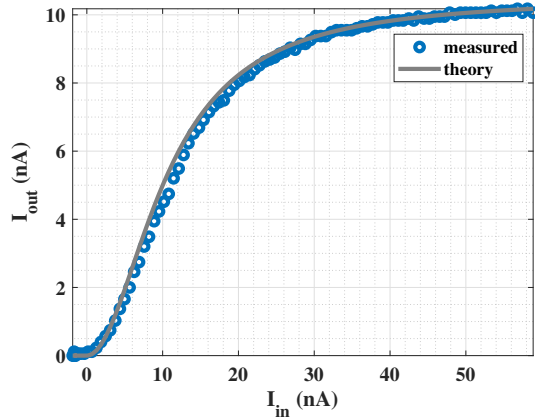


Fig. 6. Activation function transfer curve. Chip measurements of the Fig. 5 circuit closely match the theoretically-predicted behavior of Eqn. (15) for  $I_{unit} = 10.5 \text{ nA}$ . The saturating behavior is analogous to that of the original GRU's sigmoid.

Now, the current flowing through a diode-connected nMOS is unidirectional, meaning  $I_1 = \max(I_{in}, 0)$ , and we can write

$$I_{out} = I_{unit} \cdot \frac{\max(I_{in}, 0)^2}{I_{unit}^2 + \max(I_{in}, 0)^2}, \quad (15)$$

which is a dimensionalized analog of Eqn. (7). The measurement results in Fig. 6 illustrate the nonlinear, saturating behavior of this activation function.

### C. State Update

The AFUA state update, Eqn. (10), is implemented as the adaptive filter shown in Fig. 7. The currents  $I_h$ ,  $I_{\tilde{h}}$  and  $I_z$  represent the hidden state  $h_j$ , the candidate state  $\tilde{h}_j$  and the update gate,  $z_j$ , respectively. From the translinear loop principle, the Fig. 7 circuit's dynamics can be written as [32], [30]

$$\underbrace{\frac{C_z U_T}{\kappa I_{unit}}}_{\tau} \frac{I_{unit}}{I_z} \frac{dI_h}{dt} = 2I_{\tilde{h}} - I_h, \quad (16)$$

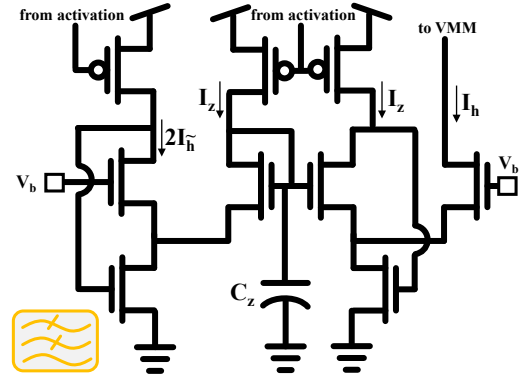


Fig. 7. State update circuit schematic. The output  $I_h$  is a low-pass-filtered version of the input,  $2I_{\tilde{h}}$ . The filter's time constant is inversely proportional to the value of the current  $I_z$ . So, large values of  $I_z$  increase the rate at which  $I_h$  updates to  $2I_{\tilde{h}}$ , while small values of  $I_z$  slow down this process.

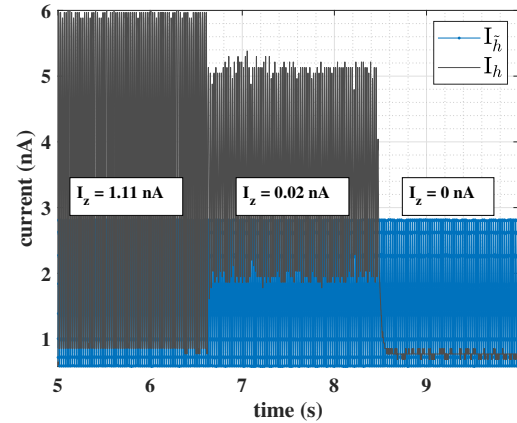


Fig. 8. State update circuit response. Chip measurements of the Fig. 7 circuit show that the output,  $I_h$ , follows the input,  $I_{\tilde{h}}$  at a rate that is determined by the value of current  $I_z$ .

where  $\kappa$  is the body-effect coefficient and  $U_T$  is the thermal voltage [33]. Just as  $z_j$  does for  $h_j$  in Eqn. (10),  $I_z$  controls the update speed of  $I_h$  (see Fig. 8).

### D. Vector Matrix Multiplication

Figure 9 depicts the components of our vector-matrix multiplication (VMM) block. These are the soma and synapse circuits that are common in the analog neuromorphic literature [34]. Crucially, the soma-synapse architecture is current-in, current-out. This means that, unlike other approaches for implementing GRU and LSTM networks [15], [16], [17], the VMM does not need power-consuming operational amplifiers to convert signals between the current and voltage domains.

## IV. ANALOG LSTM CIRCUIT ANALYSIS

The following subsections address various practical aspects of an actual AFUA implementation.

### A. Current Consumption

Since the activation function, Eqn. (7), has a range of  $(0, 1)$ , the  $z_j$  and  $\tilde{h}_j$  variables are likewise limited to  $(0, 1)$ . Also,

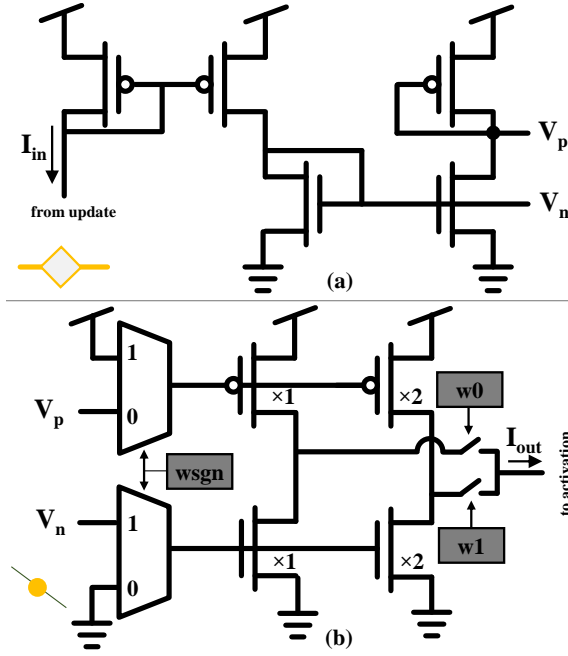


Fig. 9. Vector matrix multiplier circuit components. (a) The soma is a current-mode buffer. (b) The synapse is a programmable current mirror, with gain stored in registers  $w_{sgn}$ ,  $w_0$ ,  $w_1$ . These represent the neural network's 3-bit quantized learned weights.

from Eqn. (10),  $h_j$  spans  $(0, 2)$ . This means that all update gate and candidate state currents have a maximum value of  $I_{unit}$ , while the hidden state currents have a maximum value of  $2I_{unit}$ . With this information, we can calculate upper-bounds on the current consumption of each circuit component.

1) *Activation Function*: Not counting the input current that is supplied by the VMM, Fig. 5 shows that the only current consumed by the activation function block is the differential-pair tail current of  $I_{unit}$ . There are two activation functions per AFUA cell (one each for  $z_j$  and  $\tilde{h}_j$ ). So, for an  $m$ -unit AFUA layer, the activation function blocks draw a total current of  $m \times 2I_{unit}$ .

2) *State Update*: The total current flowing through the four branches of the state update circuit (Fig. 7) is  $2I_h + 2I_z + I_h$ , which has a worst-case value of  $6I_{unit}$ . For our  $m$ -unit AFUA network, the state update circuits consume at most  $m \times 6I_{unit}$ .

3) *VMM soma*: The soma is a current-mode buffer that drives a differential signal onto each row of the VMM (see Fig. 9). For the somas on the input and bias rows, the maximum current consumption is  $2I_{unit}$ . The somas driving the hidden state rows consume at most  $4I_{unit}$  each. So, with  $n$  inputs,  $m$  hidden states and one bias row, the somas will consume a maximum total current of  $(n + 2m + 1) \times 2I_{unit}$ .

4) *VMM core*: As depicted in Fig. 9, each multiplier element in the VMM core comprises a number of current sources that are switched on or off, depending on the values of the weight bits ( $w_{sgn}$ ,  $w_0$ ,  $w_1$ ). At worst, all current sources are switched on, in which case the VMM elements that process state variables each consume  $6I_{unit}$ , while those that process input variables or biases each consume  $3I_{unit}$ . The maximum current draw of each VMM column for an  $n$ -input AFUA layer with  $m$  hidden states is therefore  $(n + 2m + 1) \times 3I_{unit}$ . There

are  $2m$  columns, to give a total maximum VMM core current consumption of  $m(n + 2m + 1) \times 6I_{unit}$ .

5) *Total Current Consumption*: From the previous subsections, we conclude that the worst-case total current consumption of an  $n$ -input AFUA layer with  $m$  hidden states is

$$I_{tot} \leq \underbrace{(m(14 + 6(n + 2m)))}_{\text{core}} + \underbrace{(4m + 2n + 2)}_{\text{VMM soma}} \times I_{unit}, \quad (17)$$

where 'core' includes the activation function, VMM core and state update current consumption. The VMM soma is peripheral to the AFUA's operation and represents overhead cost. For instance, a 16-input, 10-unit AFUA layer would spend 3 % of its power budget as overhead.

Empirically, we found that the average current consumption of some of the AFUA blocks is significantly lower than their estimated worst-case values. In particular, the VMM consumes only  $48I_{unit}$  on average. This leads to an average AFUA total current consumption of  $62I_{unit}$ . The specific choice of  $I_{unit}$  depends on the desired operating speed, as we discuss in the following subsection.

### B. Estimated Power Efficiency

The power efficiency of neural networks is conventionally measured in operations per Watt. But this metric does not apply directly to a system like the AFUA, since it executes all of its operations continuously and simultaneously. However, we can estimate the AFUA's power efficiency by considering the performance of an equivalent discrete time system.

To arrive at the discrete-form AFUA unit, we first replace the state variables of Eqns. (8), (9) and (10) with their discrete-time counterparts. This includes the discretization  $dh_j/dt = (h_j^{(t)} - h_j^{(t-1)})/\Delta T$ , where  $\Delta T$  is the sampling period. Then, we set  $\tau = \Delta T$  to produce the following expression.

$$\begin{aligned} z_j &= f([\mathbf{W}_z \mathbf{x}]_j + [\mathbf{U}_z (\mathbf{h} - \mathbf{1}) + \mathbf{b}_z]_j) \\ \tilde{h}_j^{(t)} &= f([\mathbf{W}_x]_j + [\mathbf{U} (\mathbf{h}_{(t-1)} - \mathbf{1}) + \mathbf{b}]_j) \\ h_j^{(t)} &= z_j 2\tilde{h}_j^{(t)} - (1 - z_j)h_j^{(t-1)}. \end{aligned} \quad (18)$$

For our application,  $\mathbf{W}$ ,  $\mathbf{W}_z$  are  $2 \times 2$  matrices,  $\mathbf{U}$ ,  $\mathbf{U}_z$  are  $1 \times 2$  vectors and  $z_j$  are scalars. So, each discretized AFUA unit executes 14 multiply operations per time step. Also, there are 2 divisions due to the two activation functions (see Eqn. (7)). Not counting additions and subtractions, each discretized AFUA unit executes 16 operations per time step, to make for a total of 32 operations/step performed by the network. Assuming the sampling period of  $\Delta T = 2$  ms used in our previous eating detection systems [7], [12], this implies the AFUA performs the equivalent of 16,000 operations per second.

Now, setting  $\tau = \Delta T = 2$  ms requires a unit current of

$$I_{unit} = \frac{C_z U_T}{\kappa \tau} = 500 \cdot \frac{C_z U_T}{\kappa}, \quad (19)$$

where  $C_z = 57$  fF is the integrating capacitor of the translinear loop filter,  $U_T = 26$  mV at room temperature and  $\kappa \approx 0.42$ . This gives  $I_{unit} = 1.8$  pA. With a total current consumption of  $62I_{unit}$ , a voltage supply of 1.8 V and 16K operations per second, the AFUA's equivalent operations per Watt is 76 TOPs/W.

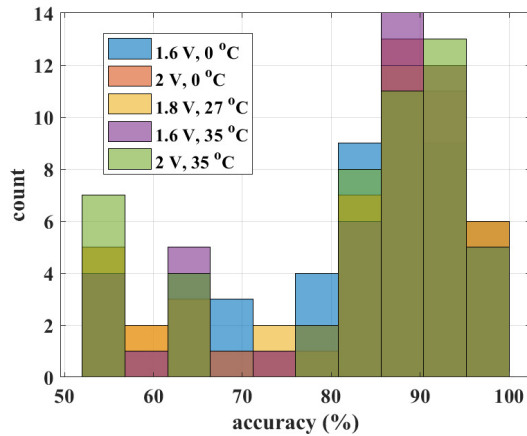


Fig. 10. Monte Carlo analysis performed for 250 runs, including mismatch and process variation, as well as power supply voltage and temperature corners of  $\{1.6V, 2V\}$  and  $\{0^\circ C, 35^\circ C\}$ , respectively. Nominal power supply voltage and temperature are  $1.8V, 27^\circ C$ . Median accuracy is 90 %.

### C. Mismatch

Due to random variations in doping and geometry, transistors that are nominally identical will exhibit mismatch when fabricated in a physical ASIC. To understand the effect of mismatch and other non-idealities on the AFUA neural network’s performance, we performed Monte Carlo analyses with foundry-provided manufacturing and test data. The Monte Carlo analyses included mismatch and process variation, as well as power supply voltage and temperature corners of  $\{1.6V, 2V\}$  and  $\{0^\circ C, 35^\circ C\}$ , respectively.

Figure 10 shows the variation in classification accuracy for 250 Monte Carlo runs of one implementation of the AFUA neural network. The median accuracy across all runs is 0.90. Most of the variation in accuracy is due to mismatch, and the AFUA neural network is largely robust to temperature, voltage and process variation. The neural network is also unaffected by circuit noise (this is a direct result of the network’s ability to generalize). To mitigate the effect of mismatch, we can use larger transistors [35], calibrate the network’s learning algorithm for each individual chip [34], or incorporate mismatch data into a fault-tolerant learning algorithm [36].

## V. EXPERIMENTAL METHODS

### A. Data Collection

Training and testing data was collected from study volunteers in a laboratory setting. All aspects of the study protocol were reviewed and approved by the Dartmouth College Institutional Review Board (Committee for the Protection of Human Subjects-Dartmouth; Protocol Number: 00030005).

The data used for this study was previously collected in a controlled laboratory setting from 20 participants (8 females, 12 males; aged 21-30) that were instructed to perform both eating and non-eating-related activities. During these activities, a contact microphone (see Fig. 11) was secured behind the ear with a headband, to measure any acoustic signals present at the tip of the mastoid bone [6]. The output of the contact microphone was digitized and stored using a 20 kSa/s, 24-bit data acquisition device (DAQ).

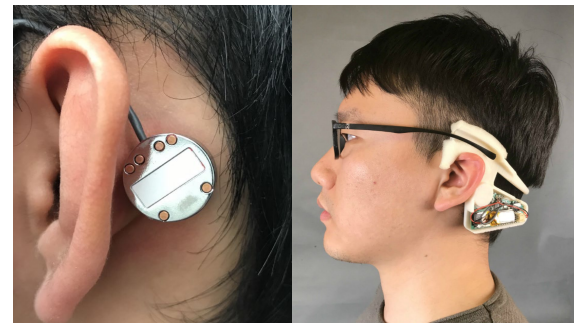


Fig. 11. Left panel: a contact microphone was used to collect acoustic data from the mastoid bone as study participants performed various eating and non-eating tasks [6]. Right panel: prototype of the complete wearable device that we are developing for dietary monitoring [7].

Participants were asked to eat a variety of foods—including carrots, protein bars, crackers, canned fruit, instant food, and yogurt—for at least 2 minutes per food type. This resulted in a 4 hour total eating dataset. Non-eating activities included talking and silence for 5 minutes each and then coughing, laughing, drinking water, sniffing, and deep breathing for 24 seconds each. This resulted in 4 hours total of non-eating data. Each activity occurred separately and was classified based on activity type as eating or non-eating.

We down-sampled the DAQ data to 500 Hz and applied a high pass filter with a 20 Hz cutoff frequency to attenuate noise. We segmented the positive class data (chewing), and negative class data (not chewing) into 24-second windows with no overlap. The positive and negative class data were labelled with the one-hot encoding (2,0) and (0,2), respectively. Finally, we extracted the ZCR-RMS and ZCR-ZCR features of the windows to produce 2-dimensional input vectors to be processed by the AFUA network.

### B. Neural Network Training

For training, the AFUA neural network was implemented in Python, using a custom layer defined by the discretized system of Eqn. (18). Chip-specific parameters were extracted for each neuron and incorporated into the custom layers. The AFUA network was trained and validated on the laboratory data (train/valid/test split: 68/12/20) using the TensorFlow Keras v2.0 package. Training was performed with the ADAM optimizer [37] and a weighted binary cross-entropy loss function to learn full-precision weights. Since the training data had a much higher sampling rate (500 Sa/s) than the bandwidth of the acoustic signals of interest (20 Hz), there was negligible information lost in the training process.

Python training was followed by a quantization step that converted the full-precision weights to signed 3-bit values (0,  $\pm 1$ ,  $\pm 2$ ,  $\pm 3$ ). An alternative approach would have been to directly incorporate the quantization process into the network’s computational graph [13]. However, we found that such an approach only slows down training with no improvement in our network’s classification performance.

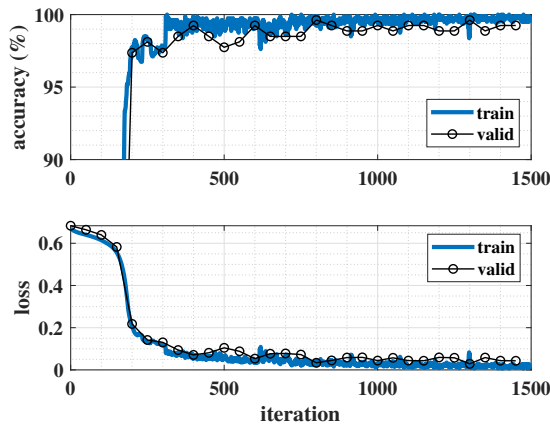


Fig. 12. Accuracy and loss training graphs for discretized AFUA neural network. We performed training in Python using the TensorFlow Keras v2.0 package. Validation set performance tracked that of the training set, indicating good generalization. The learned weights were quantized and programmed onto the AFUA ASIC’s on-chip registers.

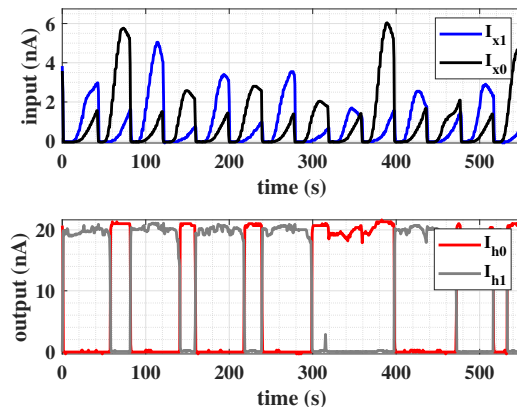


Fig. 14. AFUA chip measurement response to different input patterns ( $I_{x1}, I_{x0}$ ) taken from the test dataset.  $I_{x0}$  is the output of the cascade of an RMS block and ZCR block.  $I_{x1}$  is the output of the cascade of two ZCR blocks. The circuit’s class prediction is encoded as output currents ( $I_{h1}, I_{h0}$ ).

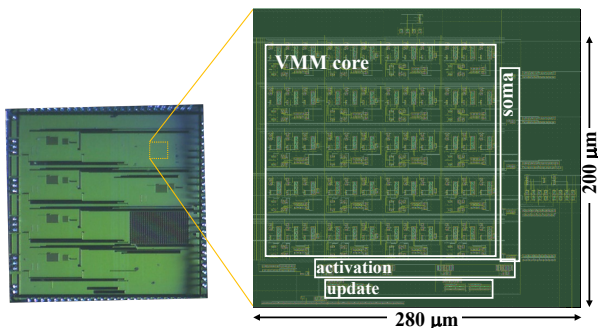


Fig. 13. Die photo of the AFUA ASIC, implemented in a  $0.18 \mu\text{m}$  CMOS process. The synapse circuits (labelled ‘VMM core’) consume most of the  $200 \mu\text{m} \times 280 \mu\text{m}$  circuit area.

### C. Chip Measurements

The AFUA was implemented, fabricated and tested as an integrated circuit in a standard  $0.18 \mu\text{m}$  mixed-signal CMOS process with a 1.8 V power supply. To simplify the measurement process and associated instrumentation, the ASIC I/O infrastructure includes current buffers that scale input currents by 1/100 and that multiply output currents by 100.

The AFUA neural network was programmed by storing the 3-bit version of each learned weight onto its corresponding on-chip register in the VMM array.

The network was then evaluated on the test dataset. Specifically, each 24-second long window of 2-dimensional feature vectors from the test dataset was dimensionalized and scaled to  $100 \times I_{\text{unit}}$  and input to the ASIC with an arbitrary waveform generator. We set  $I_{\text{unit}} \approx 10 \text{ nA}$  with an off-chip resistor. According to Eqn. (19), this  $I_{\text{unit}}$  creates a time constant of  $\tau = 0.36 \mu\text{s}$ , allowing for faster-than-real-time chip measurements—an important consideration, given the large amount of test data to be processed.

Output currents  $I_{h0}, I_{h1}$  were each measured from the voltage drop across an off-chip sense resistor. The ASIC’s steady-state response was then taken as the classification decision. An output value of  $(I_{h1}, I_{h0}) = (2I_{\text{unit}}, 0)$  means that the circuit

classified the input as eating, while  $(I_{h1}, I_{h0}) = (0, 2I_{\text{unit}})$  corresponds to non-eating. From these measurements, we calculated the algorithm’s test accuracy, loss, precision, recall, and F1-score.

## VI. RESULTS AND DISCUSSION

### A. Classification Performance

Figure 14 shows the AFUA chip’s typical response to input data. The input currents  $I_{x1}, I_{x0}$  represent the ZCR-RMS and ZCR-ZCR features extracted from the contact microphone signal. Inputting a stream of  $I_{x1}, I_{x0}$  patterns produces output currents  $I_{h1}, I_{h0}$ , which represent the hidden states of the AFUA neural network.

According to our encoding scheme,  $(I_{h1}, I_{h0}) = (2I_{\text{unit}}, 0)$  means that the circuit classified the input as chewing, while  $(I_{h1}, I_{h0}) = (0, 2I_{\text{unit}})$  corresponds to a prediction of not chewing. But the presence of noise and circuit non-ideality produces some ambiguity in the encoding: some AFUA output patterns can be interpreted as either chewing or not chewing, depending on the choice of threshold used to distinguish between 0 A and  $2I_{\text{unit}}$ . Figure 15 is the receiver operating characteristic curve (ROC) produced by varying this threshold current. The highlighted point on the ROC is a representative operating point, where the classifier produced a sensitivity of 0.91 and a specificity of 0.96. This corresponds to a false alarm rate of  $(1 - \text{specificity}) = 0.039$ .

### B. System-level Considerations

In this section, we consider the impact of using the AFUA neural network in a complete eating event detection system. To process a 500 Hz signal, the ZCR and RMS feature extraction blocks consume a total of  $0.68 \mu\text{W}$  [20]. Also, the AFUA network consumes  $1.1 \mu\text{W}$ , assuming  $I_{\text{unit}} = 10 \text{ nA}$ . Finally, a microcontroller from the MSP430x series (Texas Instruments Inc., Dallas, TX) running at 1 MHz consumes  $180 \mu\text{W}$  when active and  $0.72 \mu\text{W}$  when in standby mode [40].

The feature extraction and AFUA circuitry are always on, while the microcontroller remains in standby mode until a

	Window Size (s)	Accuracy	F1-Score	Precision	Recall	Power (mW)
<b>This work</b>	24	0.94	0.94	0.96	0.91	0.019
<b>FitByte [38]</b>	5	-	-	0.83	0.94	105
<b>TinyEats [12]</b>	4	0.95	0.95	0.95	0.95	40
<b>Auracle [6]</b>	3	0.91	-	0.95	0.87	OFFLINE
<b>EarBit [4]</b>	35	0.90	0.91	0.87	0.96	OFFLINE
<b>AXL [5]</b>	20	-	0.91	0.87	0.95	OFFLINE

TABLE II

COMPARISON BETWEEN PROPOSED EATING DETECTION SYSTEM AND PREVIOUS SOLUTIONS. THREE OF THE CLASSIFICATION ALGORITHMS [6], [4], [5] WERE IMPLEMENTED OFFLINE; SINCE THESE ARE NOT EMBEDDED SOLUTIONS, THEIR POWER CONSUMPTION IS NOT REPORTED.

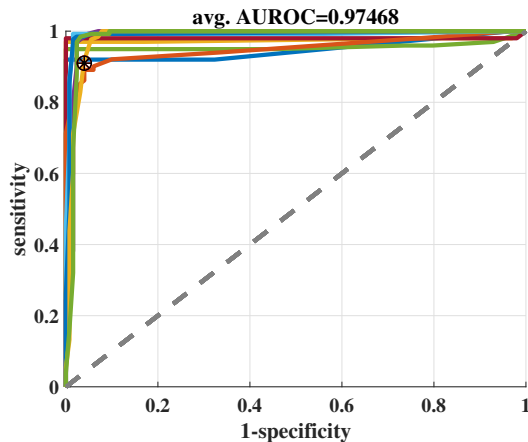


Fig. 15. Receiver operating characteristic curve (ROC) from AFUA chip measurements. These results were produced from repeated AFUA chip measurement responses to 1.6 hours of previously-unseen test data. Circuit noise produces slightly different performance from one measurement to another, with the area under ROC (AUROC) ranging from 0.95 to 0.99 (average AUROC=0.97). The highlighted point corresponds to a sensitivity of 0.91 and a specificity of 0.96.

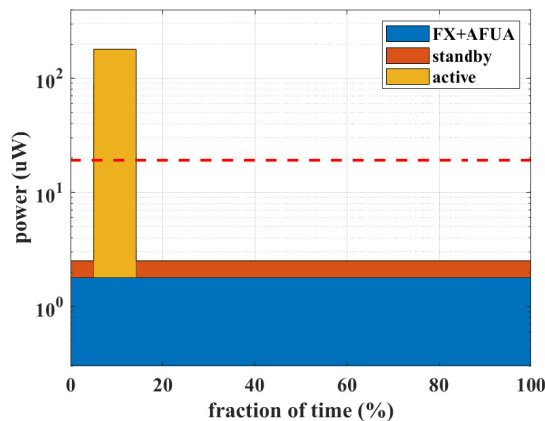


Fig. 16. Power consumption of eating detection system. The feature extraction and AFUA circuitry continuously consume  $1.8 \mu\text{W}$  of power. The microcontroller is active for 9% of the time, during which it consumes  $180 \mu\text{W}$  of power. For the remaining 91% of the time, the microcontroller consumes  $0.72 \mu\text{W}$  while in standby mode. On average (red dashed line), the whole system consumes an estimated  $18.8 \mu\text{W}$ .

potential chewing event is detected. The fraction of time the microcontroller is in the active mode depends on how often the user eats, as well as the sensitivity and specificity of the AFUA network. Assuming the user spends 6% of the day eating [39], then, using the classifier operating point highlighted in Fig. 15, the fraction of time that the microcontroller is active is

$$\begin{aligned}
 \text{ACTIVE} &= \text{EAT} \times \text{SENS} + (1 - \text{SPEC}) \times (1 - \text{EAT}) \\
 &= 0.06 \times 0.91 + (1 - 0.96) \times (1 - 0.06) \\
 &= 0.09.
 \end{aligned} \tag{20}$$

So, the microcontroller consumes an average of  $180 \mu\text{W} \times 0.09 + 0.72 \mu\text{W} \times (1 - 0.09) = 16.9 \mu\text{W}$ . As Fig. 16 shows, the average power consumption of the complete AFUA-based eating detection system is  $18.8 \mu\text{W}$ .

Table II compares our work to other recent eating detection solutions. The different approaches all yield generally the same level of classification accuracy, but our work differs in one critical aspect: while others depend on offline processing, or on tens of milliWatts of power to operate, our approach only requires an estimated  $18.8 \mu\text{W}$ .

### C. Analog versus Digital LSTM

The AFUA neural network has a total power consumption of  $1.1 \mu\text{W}$ . Unlike a digital LSTM implementation, the AFUA network is an analog circuit and does not require a front-end ADC. If we attempted to implement the system with a digital LSTM [41], [42], then it would require a 12-bit, 500 Sa/s front-end ADC [7], [43] and this ADC alone would consume over  $3 \mu\text{W}$  of power [44]. Note, the ADC would itself require an ADC driver, which typically consumes even more power than the ADC [45]; any power efficiency benefits of a digital LSTM are overwhelmed by the power demands of the ADC and ADC driver.

## VII. CONCLUSION

We have introduced the AFUA—an adaptive filter unit for analog long short-term memory—as part of an eating event detection system. Measurement results of the AFUA implemented in a  $0.18 \mu\text{m}$  CMOS technology showed that it can identify chewing events at a 24-second time resolution with a recall of 91% and an F1-score of 94%, while consuming  $1.1 \mu\text{W}$  of power. The AFUA precludes the need for an analog-to-digital converter, and it also prevents a downstream microcontroller from unnecessarily processing irrelevant data. If a signal processing system were built around the AFUA for detecting eating episodes (that is, meals and snacks), then the



whole system would consume less than 20  $\mu$ W of power. This opens up the possibility of unobtrusive, batteryless wearable devices that can be used for long-term monitoring of dietary habits.

### VIII. ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation, under award numbers CNS-1565269 and CNS-1835983. The views and conclusions contained in this document are those of the authors and do not necessarily represent the official policies, either expressed or implied, of the sponsors.

### REFERENCES

- [1] Kang, K. Nutritional counseling for obese children with obesity-related metabolic abnormalities in Korea. *Pediatric Gastroenterology, Hepatology & Nutrition*. **20**, 71-78 (2017)
- [2] O'Connor, L., Lentjes, M., Luben, R., Khaw, K., Wareham, N. & Forouhi, N. Dietary dairy product intake and incident type 2 diabetes: a prospective study using dietary data from a 7-day food diary. *Diabetologia*. **57**, 909-917 (2014)
- [3] Turton, R., Nazar, B., Burgess, E., Lawrence, N., Cardi, V., Treasure, J. & Hirsch, C. To go or not to go: A proof of concept study testing food-specific inhibition training for women with eating and weight disorders. *European Eating Disorders Review*. **26**, 11-21 (2018)
- [4] Bedri, A., Li, R., Haynes, M., Kosaraju, R., Grover, I., Prioleau, T., Beh, M., Goel, M., Starner, T. & Abowd, G. EarBit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings Of The ACM On Interactive, Mobile, Wearable And Ubiquitous Technologies*. **1**, 1-20 (2017)
- [5] Farooq, M. & Sazonov, E. Accelerometer-based detection of food intake in free-living individuals. *IEEE Sensors Journal*. **18**, 3752-3758 (2018)
- [6] Bi, S., Wang, T., Davenport, E., Peterson, R., Halter, R., Sorber, J. & Kotz, D. Toward a wearable sensor for eating detection. *Proceedings Of The 2017 Workshop On Wearable Systems And Applications*. pp. 17-22 (2017)
- [7] Bi, S., Wang, T., Tobias, N., Nordrum, J., Wang, S., Halvorsen, G., Sen, S., Peterson, R., Odame, K., Caine, K. & Others Auracle: Detecting eating episodes with an ear-mounted sensor. *Proceedings Of The ACM On Interactive, Mobile, Wearable And Ubiquitous Technologies*. **2**, 1-27 (2018)
- [8] Canhoto, A. & Arp, S. Exploring the factors that support adoption and sustained use of health and fitness wearables. *Journal Of Marketing Management*. **33**, 32-60 (2017)
- [9] Gao, Y., Li, H. & Luo, Y. An empirical study of wearable technology acceptance in healthcare. *Industrial Management & Data Systems*. (2015)
- [10] Dunne, L., Profita, H., Zeagler, C., Clawson, J., Gilliland, S., Do, E. & Budd, J. The social comfort of wearable technology and gestural interaction. *2014 36th Annual International Conference Of The IEEE Engineering In Medicine And Biology Society*. pp. 4159-4162 (2014)
- [11] Hensel, B., Demiris, G. & Courtney, K. Defining obtrusiveness in home telehealth technologies: A conceptual framework. *Journal Of The American Medical Informatics Association*. **13**, 428-431 (2006)
- [12] Nyamukuru, M. & Odame, K. Tiny Eats: Eating Detection on a Microcontroller. *2020 IEEE Second Workshop On Machine Learning On Edge In Sensor Systems (SenSys-ML)*. pp. 19-23 (2020)
- [13] Amoh, J. & Odame, K. An optimized recurrent unit for ultra-low-power keyword spotting. *Proceedings Of The ACM On Interactive, Mobile, Wearable And Ubiquitous Technologies*. **3**, 1-17 (2019)
- [14] Jordan, I. & Park, I. Birhythmic analog circuit maze: a nonlinear neurostimulation testbed. *Entropy*. **22**, 537 (2020)
- [15] Adam, K., Smagulova, K. & James, A. Memristive LSTM network hardware architecture for time-series predictive modeling problems. *2018 IEEE Asia Pacific Conference On Circuits And Systems (APCCAS)*. pp. 459-462 (2018)
- [16] Krestinskaya, O., Salama, K. & James, A. Learning in memristive neural network architectures using analog backpropagation circuits. *IEEE Transactions On Circuits And Systems I: Regular Papers*. **66**, 719-732 (2018)
- [17] Han, J., Liu, H., Wang, M., Li, Z. & Zhang, Y. ERA-LSTM: An efficient ReRAM-based architecture for long short-term memory. *IEEE Transactions On Parallel And Distributed Systems*. **31**, 1328-1342 (2019)
- [18] Zhao, Z., Srivastava, A., Peng, L. & Chen, Q. Long short-term memory network design for analog computing. *ACM Journal On Emerging Technologies In Computing Systems (JETC)*. **15**, 1-27 (2019)
- [19] Li, Q., Liu, C., Dong, P., Zhang, Y., Li, T., Lin, S., Yang, M., Qiao, F., Wang, Y., Luo, L. & Others NS-FDN: Near-Sensor Processing Architecture of Feature-Configurable Distributed Network for Beyond-Real-Time Always-on Keyword Spotting. *IEEE Transactions On Circuits And Systems I: Regular Papers*. (2021)
- [20] Baker, M., Zhak, S. & Sarpeshkar, R. A micropower envelope detector for audio applications [hearing aid applications]. *Proceedings Of The 2003 International Symposium On Circuits And Systems, 2003. ISCAS'03..* 5 pp. V-V (2003)
- [21] Sarpeshkar, R., Baker, M., Salthouse, C., Sit, J., Turicchia, L. & Zhak, S. An analog bionic ear processor with zero-crossing detection. *ISSCC. 2005 IEEE International Digest Of Technical Papers. Solid-State Circuits Conference, 2005..* pp. 78-79 (2005)
- [22] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Computation*. **9**, 1735-1780 (1997)
- [23] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv Preprint ArXiv:1406.1078*. (2014)
- [24] Zhou, G., Wu, J., Zhang, C. & Zhou, Z. Minimal gated unit for recurrent neural networks. *International Journal Of Automation And Computing*. **13**, 226-234 (2016)
- [25] Ravanelli, M., Brakel, P., Omologo, M. & Bengio, Y. Improving speech recognition by revising gated recurrent units. *ArXiv Preprint ArXiv:1710.00641*. (2017)
- [26] Ravanelli, M. & Bengio, Y. Speaker recognition from raw waveform with sincnet. *2018 IEEE Spoken Language Technology Workshop (SLT)*. pp. 1021-1028 (2018)
- [27] Strogatz, S. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. (CRC press,2018)
- [28] Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *ArXiv Preprint ArXiv:1804.03209*. (2018)
- [29] Odame, K. & Nyamukuru, M. Analog LSTM for Keyword Spotting. *2022 IEEE 4th International Conference On Artificial Intelligence Circuits And Systems (AICAS)*. (2022)
- [30] Odame, K. & Minch, B. The translinear principle: a general framework for implementing chaotic oscillators. *International Journal Of Bifurcation And Chaos*. **15**, 2559-2568 (2005)
- [31] Odame, K. & Minch, B. Implementing the Lorenz oscillator with translinear elements. *Analog Integrated Circuits And Signal Processing*. **59**, 31-41 (2009)
- [32] Mulder, J., Serdijn, W., Woerd, A. & Van Roermund, A. Dynamic translinear RMS-DC converter. *Electronics Letters*. **32**, 2067-2068 (1996)
- [33] Enz, C., Krummenacher, F. & Vittoz, E. An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. *Analog Integr. Circuits Signal Process..* **8**, 83-114 (1995)
- [34] Binas, J., Neil, D., Indiveri, G., Liu, S. & Pfeiffer, M. Precise deep neural network computation on imprecise low-power analog hardware. *ArXiv Preprint ArXiv:1606.07786*. (2016)
- [35] Pelgrom, M., Duinmaijer, A. & Welbers, A. Matching properties of MOS transistors. *IEEE Journal Of Solid-state Circuits*. **24**, 1433-1439 (1989)
- [36] Orgenci, A., Dundar, G. & Balkur, S. Fault-tolerant training of neural networks in the presence of MOS transistor mismatches. *IEEE Transactions On Circuits And Systems II: Analog And Digital Signal Processing*. **48**, 272-281 (2001)
- [37] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*. (2014)
- [38] Bedri, A., Li, D., Khurana, R., Bhuwalka, K. & Goel, M. Fitbyte: Automatic diet monitoring in unconstrained situations using multimodal sensing on eyeglasses. *Proceedings Of The 2020 CHI Conference On Human Factors In Computing Systems*. pp. 1-12 (2020)
- [39] Stimpson, J., Langellier, B. & Wilson, F. Peer Reviewed: Time Spent Eating, by Immigrant Status, Race/Ethnicity, and Length of Residence in the United States. *Preventing Chronic Disease*. **17** (2020)
- [40] TI MSP430FR596x, MSP430FR594x Mixed-Signal Microcontrollers datasheet. (Texas Instruments,2018,8), (Rev. G)
- [41] Shin, D., Lee, J., Lee, J. & Yoo, H. 14.2 DNPU: An 8.1 TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural

networks. *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. pp. 240-241 (2017)

- [42] Giraldo, J. & Verhelst, M. Laika: A 5uW programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS. *ESSCIRC 2018-IEEE 44th European Solid State Circuits Conference (ESSCIRC)*. pp. 166-169 (2018)
- [43] TI CC2640R2F Datasheet. (Texas Instruments,2020,9)
- [44] Song, S., Konijnenburg, M., Van Wegberg, R., Xu, J., Ha, H., Sijbers, W., Stanzione, S., Biswas, D., Breeschoten, A., Vis, P. & Others A 769  $\mu$ W battery-powered single-chip SoC with BLE for multi-modal vital sign monitoring health patches. *IEEE Transactions On Biomedical Circuits And Systems*. **13**, 1506-1517 (2019)
- [45] Takhti, M. & Odame, K. A power adaptive, 1.22-pW/Hz, 10-MHz read-out front-end for bio-impedance measurement. *IEEE Transactions On Biomedical Circuits And Systems*. **13**, 725-734 (2019)



**David Kotz** is the Provost, and the Pat and John Rosenwald Professor in the Department of Computer Science, at Dartmouth College. His current research involves security and privacy in smart homes, and wireless networks. He is an ACM Fellow, an IEEE Fellow, a 2008 Fulbright Fellow to India, a 2019 Visiting Professor at ETH Zürich, and an elected member of Phi Beta Kappa. He received his AB in Computer Science and Physics from Dartmouth in 1986, and his PhD in Computer Science from Duke University in 1991.



**Kofi Odame** (S'06, M'08, SM'15) is an Associate Professor of Electrical Engineering at the Thayer School of Engineering at Dartmouth College. Kofi's primary interest is in analog integrated circuits for nonlinear signal processing. This work has applications in low-power electronics for implantable and wearable biomedical devices, as well as in autonomous sensor systems.



**Maria Nyamukuru** (S'20) is currently a PhD candidate at the Thayer School of Engineering, Dartmouth College. Her research focuses on design, optimization and implementation of deep learning algorithms in resource constrained environments, with a focus on biomedical applications.



**Mohsen Shahghasemi** (S'17) received his BSc in electronics engineering from University of Zanjan in 2010 and his MSc in Electronics Engineering (Microelectronics) from Amirkabir University of Technology (Tehran Polytechnic) in 2013. From 2017 to 2022 he was at Dartmouth College where he conducted PhD research on circuit and system design for electrical impedance tomography. Mohsen is currently an Analog/Mixed-Signal Design Engineer at Apple Inc. (Irvine, CA).



**Shengjie Bi** received his BSc in Electronics Engineering from University of Electronics Science and Technology of China in 2014 and his MSc in Electrical Engineering from University of California, Los Angeles in 2016. He received his PhD in Computer Science from Dartmouth College in 2021 for work on wearable devices for automatic dietary monitoring. Shengjie is currently a Research Scientist at Meta Platforms, Inc. (Bellevue, WA).