

Power-Efficient Analog Hardware Architecture of the Learning Vector Quantization Algorithm for Brain Tumor Classification

Vassilis Alimisis¹, Graduate Student Member, IEEE, Emmanouil Anastasios Serlis², Member, IEEE, Andreas Papathanasiou, Student Member, IEEE, Nikolaos P. Eleftheriou², Member, IEEE, and Paul P. Sotiriadis², Fellow, IEEE

Abstract—This study introduces a design methodology pertaining to analog hardware architecture for the implementation of the learning vector quantization (LVQ) algorithm. It consists of three main approaches that are separated based on the distance calculation circuit (DCC) and, more specifically; Euclidean distance, Sigmoid function, and Squarer circuits. The main building blocks of each approach are the DCC and the current comparator (CC). The operational principles of the architecture are extensively elucidated and put into practice through a power-efficient configuration (operating less than 650 nW) within a low-voltage setup (0.6 V). Each specific implementation is tested on a brain tumor classification task achieving more than 96.00% classification accuracy. The designs are realized using a 90-nm CMOS process and developed utilizing the Cadence IC Suite for both schematic and physical design. Through a comparative analysis of postlayout simulation outcomes with an equivalent software-based classifier and related works, the accuracy of the applied modeling and design methodologies is validated.

Index Terms—Brain tumor dataset, current comparator (CC), distance calculation circuit (DCC), learning vector quantization (LVQ) algorithm, low-power architectures.

I. INTRODUCTION

THE convergence of machine learning (ML) and bioengineering has ushered in a new era in medical diagnosis, offering innovative solutions to enhance accuracy, efficiency, and personalization [1]. In the realm of healthcare, the synergy between ML algorithms and bioengineering techniques holds immense potential for advancing diagnostic capabilities [2]. ML excels in analyzing vast and complex biological datasets, discerning patterns, and extracting meaningful insights [3].

Manuscript received 29 December 2023; revised 19 April 2024, 14 June 2024, and 31 July 2024; accepted 14 August 2024. Date of publication 30 August 2024; date of current version 24 October 2024. This work was supported by the National Recovery and Resilience Plan Greece 2.0 funded by European Union under the NextGenerationEU Program under Project MIS 5154714. (Corresponding author: Vassilis Alimisis.)

Vassilis Alimisis and Paul P. Sotiriadis are with the Department of Electrical and Computer Engineering, National Technical University of Athens, 157 80 Athens, Greece, and also with Archimedes/Athens RC, 151 25 Marousi, Greece (e-mail: alimisisv@gmail.com).

Emmanouil Anastasios Serlis, Andreas Papathanasiou, and Nikolaos P. Eleftheriou are with the Department of Electrical and Computer Engineering, National Technical University of Athens, 157 80 Athens, Greece.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TVLSI.2024.3447903>.

Digital Object Identifier 10.1109/TVLSI.2024.3447903

When integrated into bioengineering applications, such as medical imaging, sensor technologies, and diagnostic assays, ML facilitates more precise and rapid detection of various medical conditions [4]. This collaborative approach not only streamlines the diagnostic process but also enables the development of tailored and patient-specific diagnostic tools. The combination of ML and bioengineering in medical diagnosis represents a paradigm shift, promising to redefine how we identify and understand diseases, ultimately leading to more effective and personalized treatment strategies [1].

Within the intersection of bioengineering and ML, the exploration of low-power concepts emerges as a pivotal aspect in revolutionizing medical diagnostic solutions [5], [6]. As the demand for portable and energy-efficient diagnostic devices grows, incorporating ML algorithms into bioengineering applications requires a keen focus on optimizing power consumption [7]. Implementing low-power concepts becomes particularly crucial in developing wearable diagnostic technologies, remote monitoring systems, and point-of-care devices [5], [6]. By harnessing energy-efficient designs and leveraging innovative power management strategies, the synergy between ML and bioengineering not only enhances diagnostic accuracy but also ensures the feasibility and sustainability of these technologies in diverse healthcare settings. The pursuit of low-power solutions in this collaborative field reflects a commitment to creating accessible, patient-centric diagnostic tools that can operate seamlessly in resource-constrained environments [8].

The incorporation of analog hardware classifiers marks a significant advancement in translating computational models into real-world bioengineering applications [9], [10]. Analog classifiers bring a unique set of advantages, seamlessly aligning with the low-power and high-efficiency requirements often crucial in bioengineering devices [11]. Their ability to process continuous signals aligns well with the inherent analog nature of many biological processes, fostering a more natural integration within bioengineering systems [9]. The combination of ML, bioengineering, and analog hardware classifiers holds the promise of creating intelligent, energy-efficient diagnostic tools that can operate with the speed and precision demanded by healthcare applications [12]. This collaborative approach not only enhances the capabilities of medical diagnostics but

also contributes to the development of advanced, bio-inspired computing systems capable of tackling complex challenges in health monitoring and personalized medicine [13].

A high-level block diagram of an analog integrated classification system is analyzed in [14]. It can be used for bioengineering applications based on the following concept. At its core lies a low-power smart sensor, designed to capture and preprocess relevant biological signals [15], [16]. This sensor feeds into a low-power analog front end, acting as the initial gateway for signal processing [17], [18]. The analog front end, in turn, interfaces with a low-power analog feature extractor [19], [20], [21], which extracts key features from the signal for subsequent analysis. An analog memory component [22] efficiently stores these features, allowing for seamless and rapid retrieval during the classification process. The heart of the system is the low-power analog classifier [10], leveraging the analog nature of signals for real-time, energy-efficient decision-making.

With a focus on meeting the power and space efficiency demands of biomedical smart sensor systems [16], [23], this study introduces a low-power analog integrated architecture (three approaches) based on a learning vector quantization (LVQ) classification algorithm. The proposed classifiers are suitable for accurate smart sensor systems operating on battery power, as they achieve high accuracy with power efficiency. The development and validation of these designs are meticulous, involving real-world brain tumor dataset [24]. Postlayout simulations, conducted using a TSMC 90-nm CMOS process through Cadence IC Suite, confirm the accuracy of the implemented architectures, substantiated through a comparison with a software-based counterpart. In addition, for a comprehensive assessment, this article includes a comparative analysis, evaluating the proposed classifier against analog-related classifiers.

The rest of this article is organized as follows. Section II offers the essential background information on the implemented classification algorithm. Moving forward to Section III, we delve into the high-level architecture of the proposed classifier and explore the transistor-level implementations of its fundamental building blocks. The training and tuning capabilities of this architecture are detailed in Section V. In Section VI, we evaluate the classifier's accuracy using two real-world datasets. Section VII conducts a comparison study with related analog classifiers, summarizing and discussing the main aspects. Finally, Section VIII serves as the conclusion of this article.

II. LEARNING VQ

The LVQ algorithm is closely intertwined with the unsupervised and learning methods of self-organizing maps (SOMs) and vector quantization (VQ) techniques [25]. While VQ and SOMs are geared toward unsupervised clustering and learning, LVQ distinguishes itself as a supervised learning approach. Furthermore, unlike SOMs, LVQ does not involve the definition of neighborhoods around the "winner" during the learning process, and it does not assume any spatial order of the codebook vectors that are attributed as representatives to each class.

LVQ serves a highly specialized purpose, primarily focused on statistical classification and recognition [26]. Its sole objective is to delineate class regions within the input data space. To achieve this, a subset of codebook vectors with similar labels is assigned to each class region. Even if the class distributions of input samples overlap at the class boundaries, the codebook vectors in algorithms such as LVQ consistently remain within their designated class regions. The quantization regions, akin to Voronoi sets in VQ [27], are demarcated by midplanes (hyperplanes) positioned between neighboring codebook vectors.

What sets LVQ apart further is its unique approach to class borders. LVQ selects specific Voronoi-like borders that serve the role of separating Voronoi sets into distinct classes, thereby achieving piecewise linear class boundary formation.

In terms of the update of the weights in the codebook vectors, there has been a wide range of closely related techniques, such as LVQ1, LVQ2, LVQ3, and OLVQ1 [25]. Due to the need for a continual approximation of the class borders even for large number of training iterations, the LVQ3 algorithm was selected as the software and parameter-tuning backbone of this work.

All in all, LVQ3—similar to the previously mentioned algorithms—assigns a predefined number of N_v codebook vectors to each of the N_c different classes and searches for the codebook vector w_i with the least euclidian distance (different distance metrics are also used) to the input sample x according to the following equation:

$$c = \underset{i \in \{1, 2, \dots, N_v\}}{\operatorname{argmin}} \|w_i - x\|. \quad (1)$$

The point where LVQ3 differentiates pertains to the update of only two codebook vectors w_i and w_j , where w_i is the closest vector that belongs to the same class as x and w_j is the closest vector that belongs to a class different than x . This update rule can be described via the equations

$$w_i(t+1) = w_i(t) + \alpha(t)[x(t) - w_i(t)] \quad (2)$$

$$w_j(t+1) = w_j(t) - \alpha(t)[x(t) - w_j(t)]. \quad (3)$$

In this case, $\alpha \in [0, 1]$ serves the role of the learning rate during the training procedure.

The aforementioned mechanism is differentiated, in case the minimum-distance representatives w_i and w_j have identical output class prediction as the ground-truth label of input x and take the following formulation:

$$w_k(t+1) = w_k(t) + \epsilon \alpha(t)[x(t) - w_j(t)], \quad k \in \{i, j\}. \quad (4)$$

It should be noted that the second part of the weights' update rule introduces a scale-down constant $\epsilon \in [0, 1]$ of the learning rate $\alpha(t)$ to avoid overfitting and exploding gradient phenomena. For the same reasoning, the algorithm imposes a certain window-based area $s = (1 - w)/(w + 1)$ of width $w \in [0, 1]$ around the midplane defined by the vectors w_i and w_j for the update mechanism to take place

$$\min\left(\frac{w_i}{w_j}, \frac{w_j}{w_i}\right) > s. \quad (5)$$

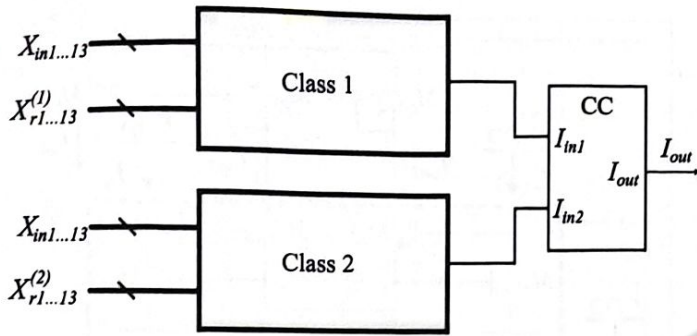


Fig. 1. Proposed classifier architecture comprises two class cells, each composed of 13 DCCs, and a CC that determines the winning class by comparing the two input currents.

III. PROPOSED ARCHITECTURES AND CIRCUITS

In this section, we analyze the proposed analog high-level architecture of the classifier, along with the basic building blocks. The high-level architecture consists of distance calculation circuits (DCCs), cascode current mirrors (CMs), and a current comparator (CC). In this work, three different DCCs are implemented, analyzed, and used as basic building blocks in the proposed architecture. The summation of the classes' output node is carried out through the cascode CMs. These are utilized in order to minimize potential distortions in the calculations that might arise from undesirable effects on the output currents of the DCCs.

A. System Analysis

The high-level architecture of the LVQ classification system regards a general-purpose problem with $N_c = 2$ classes, $N_d = 13$ feature inputs, and $N_v = 1$ codebook vectors per class and is depicted in Fig. 1 (the selection of the number of classes and features is related to the classification task). As it is shown, both classes receive the same input characteristics, with their tuning pertaining to the different weight parameters X_r (V_r or I_r). Since a single codebook vector is being utilized in this work for low-power consumption purposes, each class holds a single 13- D representative vector. Each class structure outputs a distance metric current, which could be chosen to be proportional or reversely proportional to the multidimensional Euclidean distance between X_{in} (V_{in} or I_{in}) and X_r based on (1). As a result, this capability of circuit choice leads to the necessity for modified class architecture, with a parallel topology similar to Fig. 2. It is needed when $I_{out} \propto |X_{in} - X_r|$.

Next, the class output currents are directly compared via a CC, which leads to the indication of the winning class. Based on the distance circuit utilized, CC should be a loser-take-all decision-making circuit. The aforementioned circuit comes with the advantage that the output representation is in a binary format.

In terms of potential limitations of the proposed architecture, one should mention that parallel-connected DCCs as in Fig. 2 come with the drawback of increased output current, especially for the losing classes. That could lead to increased distortion at Kirchoff's law summation on the CM between the class and the CC block. Overall, when tuned accordingly, this implementation can provide competitive results both in terms

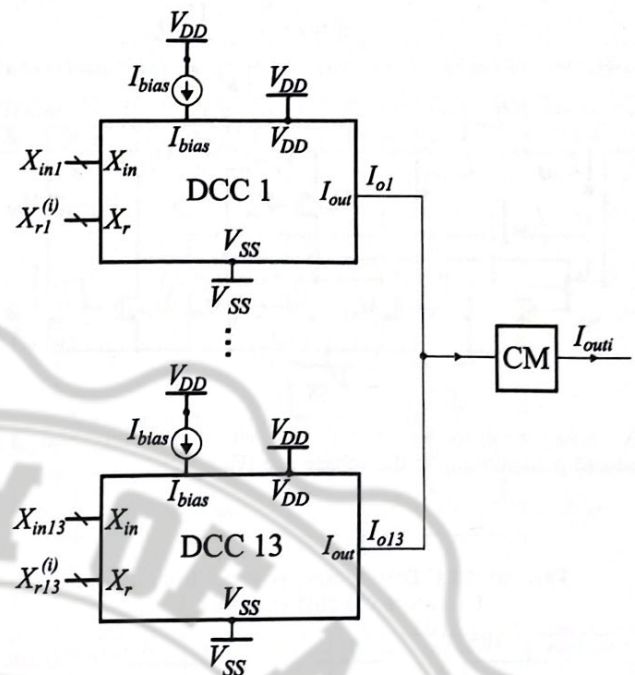


Fig. 2. Implementation of proposed classifier's class i (class 1 or class 2). It comprises 13 DCCs that describe the appropriate distance metric for each feature, along with a cascode CM used to implement the current summation on the output node.

of classification accuracy and power savings, as demonstrated in Section VI.

Given that achieving a low-power design constitutes a primary objective of this study, all transistors function within the subthreshold region, with the power supply rails established at $V_{DD} = -V_{SS} = 0.3$ V. The choice of fundamental building blocks and power supply levels is driven by a delicate balance between attaining high accuracy, minimizing power consumption, and ensuring the correct operational principles of the entire classifier. Furthermore, we conduct noise-transient simulations to assess the behavior of the proposed classifier. The classification outcome demonstrates notable resilience, suggesting that errors arising from internal noise are relatively minor compared to those stemming from data inaccuracies. We have also confirmed that noise does not degrade the accuracy and precision of circuit operation, especially in this application, as the signal levels are not close to the noise floor. The rms voltage for the signal (square pulse) is equal to $V_{sigRMS} = 35$ mV, and the integrated noise over the relevant bandwidth (BW) (1 Hz–200 KHz) is equal to $V_{noise} = 13.71$ μ Vrms (based on the simulation results).

B. Euclidean Distance Circuit

The first circuit to be examined as DCC is the one that emulates the Euclidean distance function required by the LVQ algorithm and is depicted in Fig. 3 [28]. The circuit comprises two double differential pairs that receive voltage values $X_{in} = V_{in}$ and $X_r = V_r$ as inputs.

In case the input V_{in} is higher than the mean value V_r , the left-hand side differential amplifier produces an output current I_{out1} that is proportional to the potential difference $V_{in} - V_r$. In a similar manner, when $V_r < V_{in}$ and assuming symmetrical sizing for all transistors M_{p1} – M_{p4} as shown in

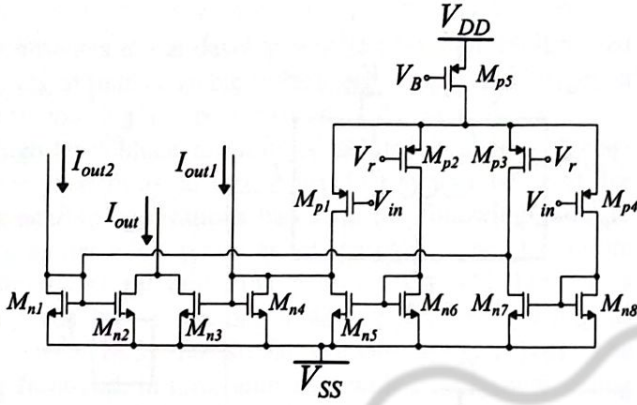


Fig. 3. Schematic of the Euclidean distance circuit. The output current I_{out} is produced proportionally to the voltage gap $|V_{in} - V_r|$.

TABLE I
TRANSISTORS' DIMENSIONS FOR THE EUCLIDEAN
DISTANCE CIRCUIT (SEE FIG. 3)

Transistors	W/L ($\mu\text{m}/\mu\text{m}$)	Transistors	W/L ($\mu\text{m}/\mu\text{m}$)
$M_{n1}-M_{n8}$	0.4/2.4	$M_{p1}-M_{p4}$	0.4/2.4
-	-	M_{p5}	2.0/2.4

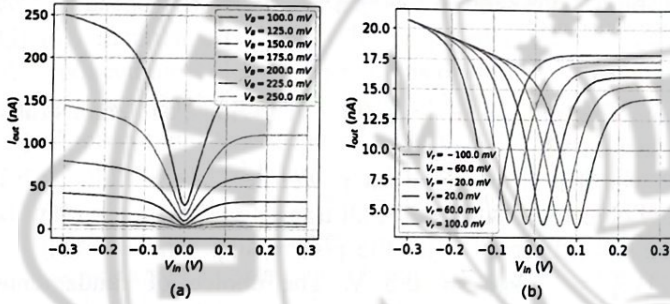


Fig. 4. (a) Output current of the Euclidean distance circuit as a function of V_{in} for a range of bias voltage values $V_B \in [150, 300 \text{ mV}]$. (b) Output current of the Euclidean distance circuit as a function of V_{in} for a range of mean voltage values $V_r \in [-100, 100 \text{ mV}]$.

Table I, an equivalent increase is noticed at the I_{out2} node, while I_{out1} currents derails to zero.

The biasing of the circuit is regulated via the W/L ratio and the gate input voltage $V_G = V_B$ at the pMOS-type transistor M_{p5} . The voltage V_B tunes the bias current I_{bias} of the circuit. In particular, an increase in V_G leads to a drop-off in output current due to a lower $|V_{GS}|$ value, as shown in the parametric analysis of Fig. 4(a).

The final output current, after correct sizing of the CM transistors $M_{n1}-M_{n4}$, is directly proportional to the output currents I_{out1} and I_{out2} and is given by the following equation:

$$I_{out} = I_{out1} + I_{out2}. \quad (6)$$

In terms of the operation region, the circuit functions adequately well over small potential differences $|V_{in} - V_r|$, with a suboptimal output current being noticed for voltage gaps that are close to the rail voltages $V_{DD} = -V_{SS} = 300 \text{ mV}$, as shown in Fig. 4(b). Thus, as described in Section V, the optimal region of functionality for the Euclidean distance circuit is set at the $[-100, 100 \text{ mV}]$ range.

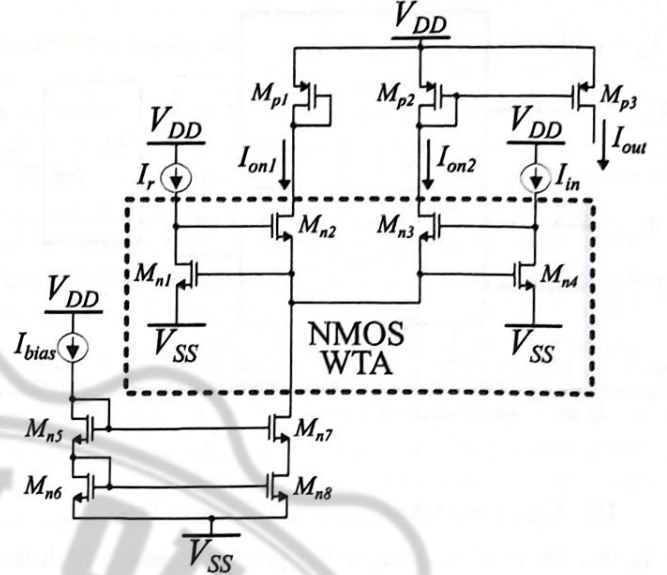


Fig. 5. High-level schematic of the proposed SFC. It consists of an nMOS cascode CM, an nMOS WTA circuit, and a pMOS CM. The I_r parameter current tunes the mean value, and I_{bias} alters the height of the Sigmoid function curve.

C. Sigmoid Function Circuit

In this section, an alternative implementation of a Sigmoid function circuit (SFC) [29] is introduced. The proposed SFC consists of three main subblocks, an nMOS cascode CM, a pMOS CM, and an nMOS winner-takes-all (WTA) circuit. The cascode CMs are used to enhance mirroring even for small bias currents. By using cascode CMs, the channel-length modulation effect (early effect) is reduced, and the quality of the mirroring is increased. Also, the cascode configuration provides some level of immunity to noise and interference, improving the signal-to-noise ratio and overall performance of the circuit. A typical SFC consists of a differential pair, which (in this implementation) is replaced by the nMOS WTA in order to achieve a sharper Sigmoid function curve. The WTA circuit is used because it provides higher linearity compared to a typical differential pair. The proposed SFC is illustrated in Fig. 5.

The circuit configurations of the nMOS WTA circuit designed for two inputs are also shown in Fig. 5. The pMOS WTA circuit serves as its symmetric counterpart and can be easily designed. It is constructed using four nMOS transistors with $(W/L) = (400/1600 \text{ nm})$. These transistors operate in the subthreshold region and are biased by a constant current denoted as I_{bias} . The functioning of the WTA circuit is illustrated in Fig. 6 [30]. In this context, I_{bias} is set at 5 nA , where I_{in1} represents a parametric current equivalent to I_{in} , and I_{in2} is fixed at 6 nA (I_r). If $I_{in1} > I_{in2}$, then $I_{on1} = I_{bias}$ and $I_{on2} = 0$. For equal input currents, then the output currents are equal too. In a different scenario, $I_{on2} = I_{bias}$ and $I_{on1} = 0$.

The electronic adjustment of the Sigmoid function's height and center is accomplished by manipulating two circuit parameters: I_{bias} and I_r . An additional parameter, V_c (associated with bulk-controlled transistors), can be introduced to fine-tune the width of the Sigmoid function although it does not impact the classification accuracy (for this circuit). These parameters

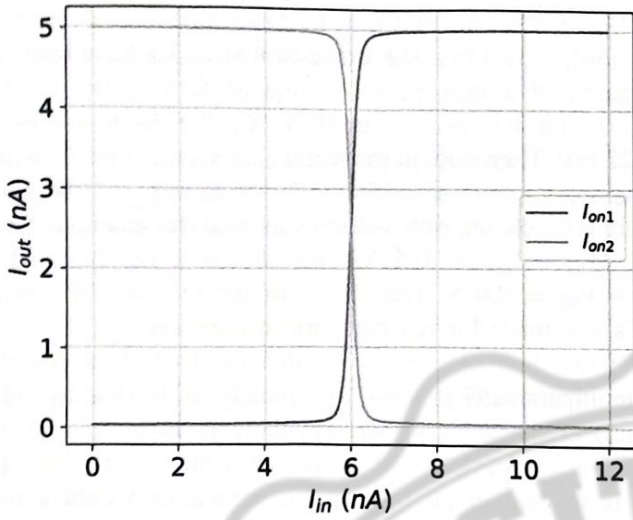


Fig. 6. In this configuration, $I_{\text{bias}} = 5$ nA, $I_{\text{in}1}$ is a parametric current equal to I_{in} , and $I_{\text{in}2} = I_r = 6$ nA. The output current for both neurons as function of the input current $I_{\text{in}1}$.

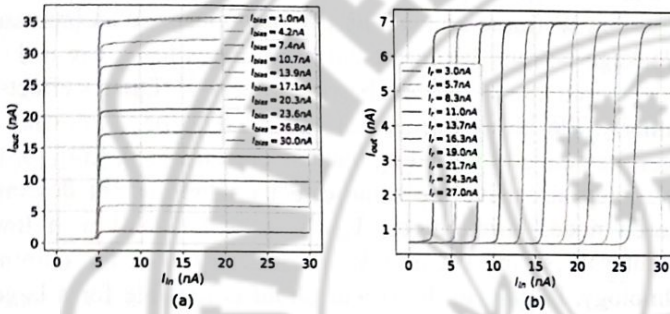


Fig. 7. (a) Output current of the SFC as a function of I_{in} and parameterized on I_{bias} , for $I_r = 5$ nA. (b) Output current of the SFC as a function of I_{in} and parameterized on I_r for $I_{\text{bias}} = 5$ nA.

are determined during the training process of the classifier, which is carried out through a software-based implementation. The bias current I_{bias} , as shown in Fig. 7, adjusts the height of the resulting Sigmoid output current while maintaining a constant $I_r = 5$ nA. The mean value of the derived Sigmoid function is modified by the current I_r , as illustrated in Fig. 7, while keeping the value of $I_{\text{bias}} = 5$ nA constant. The SFC's transistor dimensions are equal to $(W/L) = (400/1600$ nm) (for nMOS) and $(W/L) = (1600/1600$ nm) (for pMOS).

D. Squarer Circuit

The implemented Squarer circuit is shown in Fig. 8 and operates entirely in the current domain [31]. The implemented design leverages the translinear principle, which is commonly employed to efficiently realize current multiplication and division in current-mode circuits. Transistors M_{n1} , M_{n2} , M_{p1} , and M_{p2} form a translinear loop that makes up the core of the presented architecture. Analysis of this loop yields

$$I_{\text{out}} = \frac{I_{\text{in}}^2}{I_r} \quad (7)$$

The topology is expanded with CMs to buffer the input and output currents, isolating the translinear core and ensuring proper operation of the circuit. The cascode CMs are used to enhance mirroring even for small bias currents. It is also noted

TABLE II

TRANSISTORS' DIMENSIONS FOR THE SQUARER CIRCUIT (SEE FIG. 8)

Transistors	W/L ($\mu\text{m}/\mu\text{m}$)	Transistors	W/L ($\mu\text{m}/\mu\text{m}$)
M_{n1} - M_{n8}	0.8/2.4	M_{p1} - M_{p6}	0.8/2.4

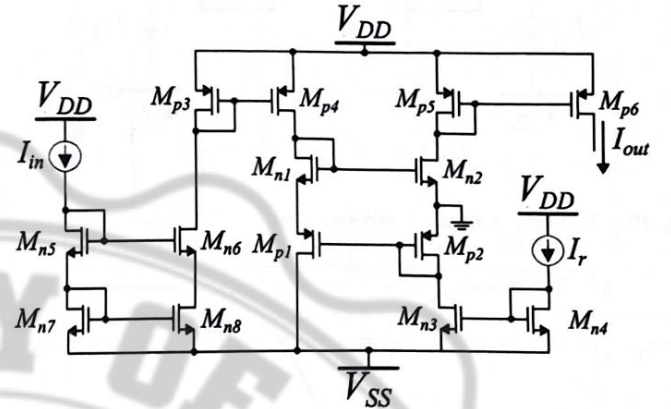


Fig. 8. Translinear circuit for computing (I_{in}^2/I_r) . It is a Squarer function circuit. Both I_{in} and I_r tune the output current.

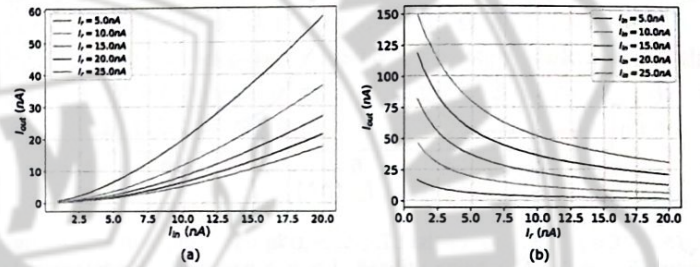


Fig. 9. (a) Output current of the Squarer circuit as a function of I_{in} and parameterized on I_r . (b) Output current of the Squarer circuit as a function of I_r and parameterized on I_{in} .

that for (7) to hold true, the transistor pairs of M_{n1} , M_{n2} and M_{p1} , M_{p2} must have the same aspect ratio, i.e., $(W/L)_{n1} = (W/L)_{n2}$ and $(W/L)_{p1} = (W/L)_{p2}$, which is also shown in Table II. The output current follows the expected behavior of (7), as shown in Fig. 9. The current I_r is used for tuning the output current, as depicted in Fig. 9.

E. Loser-Take-All Circuit

Designing and using a DCC as the key metric for calculating the association of inputs with their prototypes imply that, now, the winning class corresponds to the classification block with the smallest output current. Therefore, it becomes necessary to make use of a circuit, which will correspond to the argmin operator for the N_c different classes. Such a circuit is shown in Fig. 10 [32].

Regarding its operation, it suffices to consider the case where one of the input currents is much smaller, for instance, $I_{\text{in}1}$. This will then cause the voltage at the node $V_{\text{GM}p1}$ and $V_{\text{DM}p1}$ to rise, as the transistor M_{p1} is driven into the cutoff region. In fact, the increase in the drain is larger due to the increased impedance ($V_{\text{DM}p1}$ is connected to the gate of M_{n1}). Thus, for the nMOS transistor, it is true that the voltage $V_{\text{GSM}n1}$ increases since $V_{\text{GSM}n1} = V_{\text{DGM}p1} = V_{\text{DM}p1} - V_{\text{GM}p1} > 0$. By corresponding reasoning, if we have two classes with the

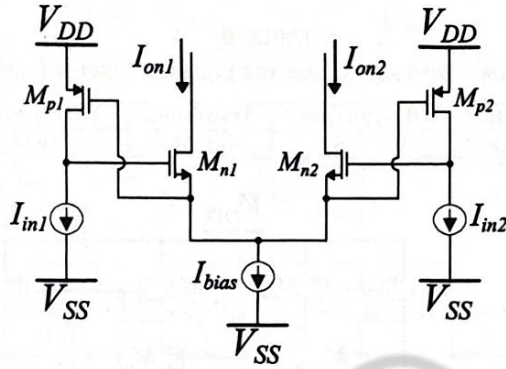


Fig. 10. Two-neuron standard LTA circuit schematic.

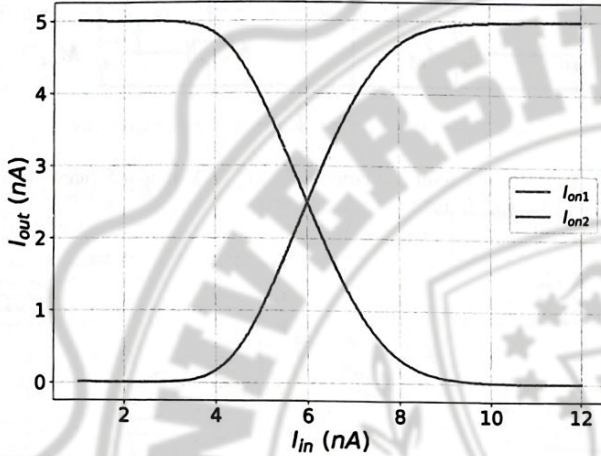


Fig. 11. Output currents of the LTA circuit. In this configuration, a biasing current $I_{bias} = 5$ nA is being utilized. I_{in1} is a parametric current equal to I_{in} and $I_{in2} = 6$ nA. The output current for both neurons is a function of the input current I_{in1} .

same input current (let $I_{in1} = I_{in2}$), then we get $I_{out1} = I_{out2} = (I_{bias}/2)$ due to the same potential difference V_{GS} in the output nMOS transistors.

The results of a two-input LTA circuit are presented in Fig. 11, where its functionality is validated. It should be mentioned that in contrast to the sharp decision boundary that the WTA circuit exhibits in Fig. 6, the LTA circuit is showing a significantly more linear region, leading to potential multiple winners' phenomena. However, since LTA is being utilized alongside parallel-connected codebook vectors, the necessity for a cascaded topology [14] is minimized due to the increased gap between the current of the losing class and the rest. As a result, two extra LTA blocks are being saved, leading to a decrease in power consumption.

IV. DESIGN PROCEDURE

In this section, we will analyze the process of selecting the specifications and design parameters for the proposed architecture. Starting from the power supply, the choice was made based on low power consumption and proper circuit operation in the subthreshold region over process–voltage–temperature (PVT) variations [33], [34], [35], [36]. Specifically, for low-power applications, the ideal operating region is subthreshold, where devices should be biased with V_{GS} voltages almost equal to V_{th} (which increases with decreasing temperature due to the increase in carrier mobility)

and $V_{DS} \geq 4V_T$ where $V_T = kT/q$ (temperature-dependent) [33], [34], [35], [36]. The implemented DCCs have branches consisting of a maximum of three or four transistors. For a maximum temperature of 125 °C, the value of V_T is 34.322 mV. Therefore, in the worst case scenario for transistor operation, $V_{DD} - V_{SS} = 549.152$ mV is required. To allow margin for reducing over voltage variation (for example, lower case: $V_{DD} - V_{SS} = 0.5$ V), we choose a supply equal to $V_{DD} - V_{SS} = 0.6$ V. The choice of the same supply for all DCCs was made for fair comparison purposes.

The selection of dimensions in each DCC is a complex and multiparameter process. Specifically, as W (width) and L (length) of the devices increase, there is an increase in the total area occupied. Based on previous implementations, the goal is to contain all three implementations within a total area smaller than 0.25 mm² based on the literature [33], [34], [35], [36]. According to both literature and simulations (for the subthreshold region), opting for a small value of W and as large as possible for L is desirable. Specifically, increasing W (more conducting channels) leads to an increase in leakage current, while increasing L tends to reduce it due to decreased drain-induced barrier lowering (DIBL) effects. Similarly, the bias currents were selected to be significantly larger than the corresponding leakage currents [33], [34], [35], [36]. They also have the same effect on the current flowing at each node. Furthermore, V_{GS} voltage is small with low current values, and because V_{th} is quite small in the current technology, biasing in the subthreshold is feasible for a large L and relatively small W . The value of V_{th} increases with both the increase in L and the decrease in W (reducing the gate-channel capacitance). Furthermore, flicker noise is reduced with the increase of L . The choice of value also depends on the desired noise level, which, based on transient simulations, is not more affected by data inaccuracies [33], [34], [35], [36]. The simulation results regarding the noise level on the output node for the three implementations are summarized in Fig. 12. In addition, with an increase in W and L values, there is an increase in parasitic capacitances, which decrease the desired BW of the classifier. This results in a reduction in its processing speed. The effect of the choice of dimensions on the input–output transistors, which create the dominant pole through the parasitic capacitors, is shown in Fig. 13, both with the alteration of W (values) and with the L (values). Our goal was above few KHz based on the literature [37]. Also, the low supply voltage of the system leads to reduced BW.

The systematic offset is heavily connected to the design process. A specific offset may be a parameter of the design. However, design faults can lead to these kinds of offsets. In order to size appropriately the architecture's transistors, Pelgrom's model is also used [38]. The variance of the Gaussian distribution, which is related to the device's parameters on the wafer, is derived as

$$\sigma^2(\Delta P) = \frac{A_p^2}{WL} + S_p^2 D^2 \quad (8)$$

where ΔP is the difference in some device parameter P between two devices spaced a distance D apart. A_p and S_p are proportionally constants obtained from experimental

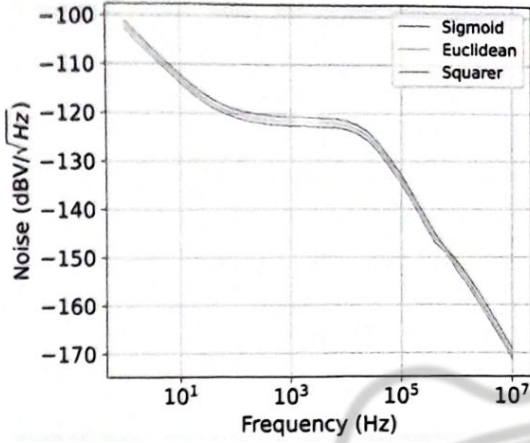


Fig. 12. Noise level for the three implementations on the output node.

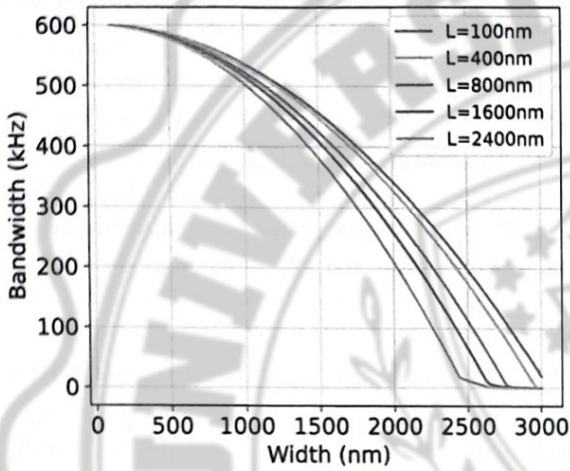


Fig. 13. Effect of the choice of dimensions on the input-output transistors, which create the dominant pole through the parasitic capacitors. In this graph, we alter both W and L .

measurements. The related equations for V_{th} and β are

$$\sigma^2(V_{th}) = \frac{A_{V_{th}}^2}{WL} + S_{V_{th}}^2 D^2 \quad (9)$$

$$\frac{\sigma^2(\beta)}{\beta^2} = \frac{A_{\beta}^2}{WL} + S_{\beta}^2 D^2. \quad (10)$$

In order to reduce the mismatch in V_{th} and β , we should increase the dimensions of transistors. Based on the previous paragraph and Pelgrom's model, we can find the optimum value for W and L [38]. Last but not least, the mismatch in a simple CM can be minimized by the area of the device and decreasing the ratio (g_m/I_D). In the weak inversion, the transconductance is given by

$$g_m = \frac{I_D}{nV_T} \quad (11)$$

where n represents the ideality factor of the transistor. Based on (11), the mismatch in a simple CM is given by

$$\frac{\sigma_{I_D}}{I_D} = \frac{1}{nV_T} \frac{A_{V_{th}}}{\sqrt{WL}}. \quad (12)$$

In comparison with strong inversion, in weak inversion, the mismatch can be reduced by increasing W or L . The selection of dimensions was made aiming for a small variation in current mirroring, less than 5% over PVT variations. The effect of a

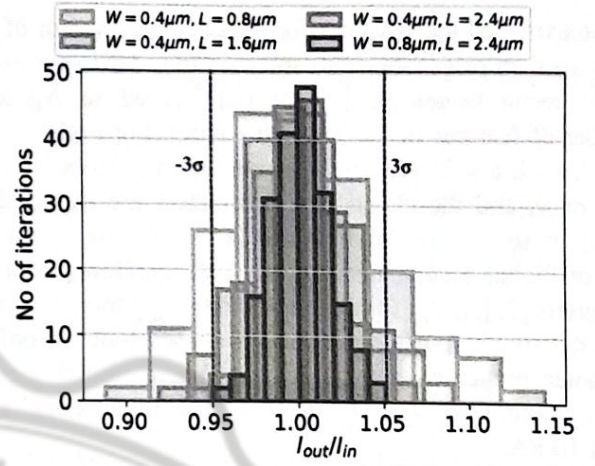


Fig. 14. Effect of mismatch for different values of W and L . For lower sizes, the output current has a large variance in comparison with the input current.

mismatch for different values of W and L is shown in Fig. 14. The larger the dimensions of the devices (differential pair or CM), the smaller the variation between the input and output current, at the cost of the total area. This is also confirmed by (12). To achieve the desired mismatch (below 5%), the product $W \cdot L$ must be greater than $1.92 \mu\text{m}^2$ (e.g., $W = 0.8 \mu\text{m}$ and $L = 2.4 \mu\text{m}$).

V. TRAINING AND TUNING CAPABILITIES

The aforementioned proposed architectures have been developed based on the premise that accurate control can be achieved via a variety of voltage and current signals. Since it consists of different DCCs, it mainly includes the mean values V_r and I_r , alongside biasing parameters for all different distance metric blocks. Such versatility is of high significance when fabricating a prototype chip since software and/or hardware variables could need to be altered in order to ensure accurate on-the-edge deployment. Furthermore, the classification system is capable of having a changeable number of inputs and classes according to the dataset provided each time for training and inference.

A. Offline Training

In terms of training, a software system that emulates the LVQ classifier is set up. The given dataset is first normalized to the voltage/current range that is within the optimal range of the designed distance metrics. In this work, the mean voltage V_r is clamped within the values of $[-100, 100]$ mV based on the results of the parametric analysis of Fig. 4(b). On the other hand, for the current-controlled Squarer circuit, the equivalent operational range was set at the $[3, 12]$ nA, while it was extended to a $[3, 15]$ nA width for the Sigmoid function due to its improved functionality for higher current values, as demonstrated in Fig. 7. In all variations, the software counterpart is being trained for a fixed number of epochs and learning rate values $\alpha(t)$ and ϵ based on (4). At the same time, according to the hardware distance circuit that is being utilized each time, the software-based distance criterion is being switched in order to minimize the difference between the software and hardware models.

The extracted mean values for the codebook vector of each class are then preprocessed in software in order to correspond to the mean values $[V_{ri}]_{i=1}^{N_d}$ or $[I_{ri}]_{i=1}^{N_d}$, where N_d is the number of features of the referred dataset. For each software model, such a software/hardware co-design process is needed only once, and the changeable parameters are exported and stored in some form of memory (either analog memristive type or digital accompanied by low rate ultralow-power data converters [22], [39]). In terms of the biasing parameters, those were determined outside the software-based loop, in order to guarantee proper circuit operation, while minimizing power consumption. Consequently, I_{bias} in all different circuits was set at 10 nA.

B. Architecture Tunability

All the proposed combinations of circuits in the system architecture can be easily adjusted for a wide range of input dimensions and the number of classes. This can be achieved via the versatile configuration of the proposed architecture in Fig. 1, where the circuit blocks used can be easily deactivated via their biasing parameters. In particular, such ON-OFF behavior is quite intuitive in the case of the Euclidean and Sigmoid circuits, which all need an appropriate I_{bias} current for their proper operation. On the other hand, switching off the Squarer distance metric can only be achieved approximately by adjusting I_{in} and I_r in the case where $I_{in} \ll I_r$, which would lead to sub-nA current output according to the parametric results in Fig. 7. An alternative implementation of the switching behavior of the various circuit blocks would require the deployment of a MOSFET switch or a transmission gate between the circuit output node and the following CM.

Based on the aforementioned techniques, a variety of different classification tasks with $n_d < N_d$ features could be tackled from a single fabricated LVQ chip with N_d initial distance metric functions per class, without the need to iterate the same design procedure proposed in this work. However, someone should consider different methodologies for such purpose for the parallel connected topology of Fig. 2, where a total of $N_d - n_d$ circuits should be cutoff via proper tuning of the I_{bias} parameters. Alternatively, similar behavior can be achieved by carefully setting X_{in} and X_r values in order to achieve negligible current output. In particular, for the Squarer and Sigmoid circuits, that would require $I_{in} \approx 0$ nA. It should be noted that cutoff of the Euclidean block is only feasible for $V_b = V_{DD} = 300$ mV since all combinations of V_{in} and V_r —for $V_b < 300$ mV—would lead to nonzero correlation output current.

Similar tuning capabilities are possible for a smaller classification problem of n_c distinct classes in contrast to the initial N_c codebook vectors fabricated. The main purpose of such a task would pertain to the minimization of the output current in the nonutilized $N_c - n_c$ codebook vectors. For parallel-connected codebook vectors as in Fig. 2, the zero current-output methods described previously for the Sigmoid, Euclidean, and Squarer circuits can be deployed for all the N_d circuits of the remaining $N_c - n_c$ codebook vectors.

Finally, the most significant adjustability of the classification chip is relevant to the electronic tunability of its different

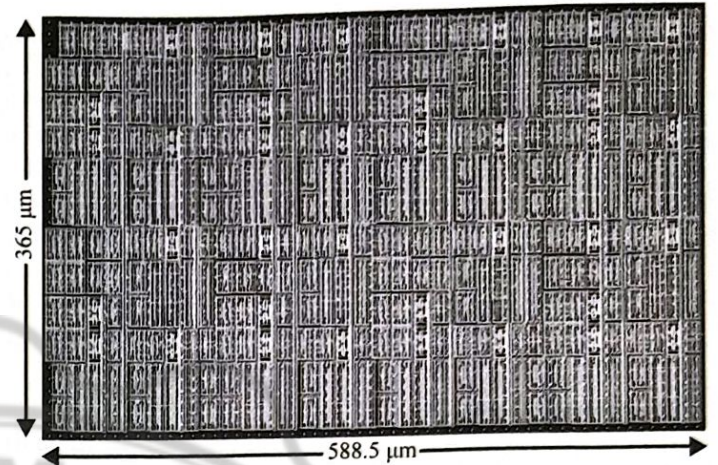


Fig. 15. Layout of the proposed classifier architectures. The total area is equal to 0.2148 mm². This layout consists of the three classifiers (more specifically: three types of DCCs, CMs, analog switches, and a CC).

parameters. That is achieved since V_r and I_r weight values are derived from a software-implementation, allowing an upgrade in performance, in case a more sophisticated LVQ algorithm is developed. Thus, the overall design is not hindered by the need for an early fabrication and deployment of the classification chip.

VI. BRAIN TUMOR DATASET AND SIMULATION RESULTS

The LVQ-based classification systems designed above were tested on a real-world dataset in order to prove the validity of the proposed methodology. A direct comparison between the software and hardware implementations is provided in order to gather knowledge about the accuracy and sensitivity of the analog classifiers. Design, simulation, and layout procedures were all conducted on the Cadence IC suite in a TSMC 90-nm CMOS process. The layout's implementation, as shown in Fig. 15, is based on the common-centroid technique, and additional dummy transistors are incorporated to mitigate mismatches and address manufacturing considerations. The total area is equal to 0.2148 mm² (the layout incorporates the three analog classifiers). All the simulation results are conducted from the layout. It combines the two classes and the CC. Each class comprises the three types of DCCs (both 13 Euclidean, 13 Sigmoid, and 13 Squarer circuits), a CM, and a switch used to select the suitable circuits. Each class consists of three 13 DCCs (Euclidean, Sigmoid, and Squarer), a CM, and a switch in order to select the appropriate circuits.

When it comes to the data acquired, the Brain Tumor feature dataset [24] that was utilized includes 1640 images that were processed for feature extraction purposes. Those refer to different attributes for the brain tumor classification task, which can be shown in Table III. In particular, 13 features were included for training and testing of the software LVQ pipeline, where five of those are first-order statistical features and the eight remaining are relevant to the image texture. The classification problem setup is binary, with the objective being the correct prediction of an existing tumor based on the features.

As a result of the dataset structure, all the implemented hardware architectures included two distinct classes for tumor

TABLE III

EXTRACTED FEATURES FROM THE BRAIN TUMOR DATASET

Feature Name	Type	Value Range
Mean	Statistical	0.07-33.23
Variance	Statistical	3.14-2910.58
Standard Deviation	Statistical	1.77-53.94
Skewness	Statistical	1.886-36.93
Kurtosis	Statistical	3.94-1371.64
Contrast	Texture	3.194-3382.57
Energy	Texture	0.02-0.58
Angular Second Moment	Texture	0-0.347
Entropy	Texture	0-0.394
Homogeneity	Texture	0.105-0.81
Dissimilarity	Texture	0.68-27.82
Correlation	Texture	0.54-0.98
Coarseness	Texture	0-1

TABLE IV

CLASSIFICATION RESULTS ON THE BRAIN TUMOR DATASET OVER 1000 ITERATIONS

	Best (%)	Worst (%)	Mean (%)
Euclidean Software	99.20	96.00	97.46
Euclidean Hardware	98.70	96.00	97.34
Sigmoid Software	99.50	96.80	98.15
Sigmoid Hardware	99.20	96.30	97.83
Squarer Software	99.70	96.60	98.17
Squarer Hardware	99.50	96.20	97.98

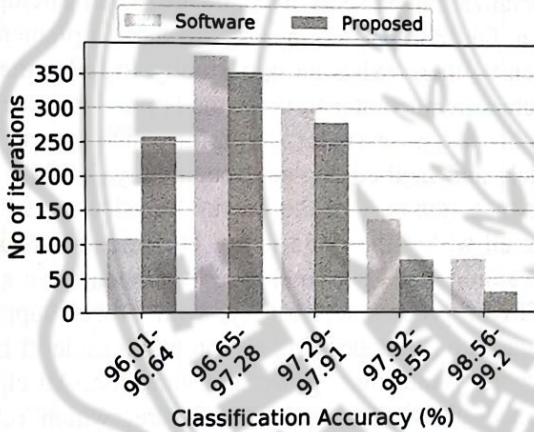


Fig. 16. Classification results of the LVQ (based on the Euclidean circuit) and the equivalent software model on the brain tumor dataset over 1000 iterations.

detection, with each class consisting of 13 distance metric circuits in a parallel way of connection. Finally, due to the abundance of data, an 80%–20% training–testing split was utilized, resulting in a test set of 329 images for validation.

In order to minimize potential overfitting phenomena, all three distinct LVQ classifiers underwent the training–testing procedure for 1000 times with the relevant results being summarized in Table IV and Figs. 16–18. More specifically, to account for random effects induced by the training algorithm, these 1000 separate software-based training iterations are conducted to extract the necessary parameters of the LVQ. As it can be clearly shown, all three implementations can achieve near-perfect accuracy results on the best train–test split. Furthermore, the software–hardware comparison validates the precision of all topologies, as they exhibit a sub-1% decrease between mean software and hardware performance. Note that there is a slight decrease in hardware accuracy

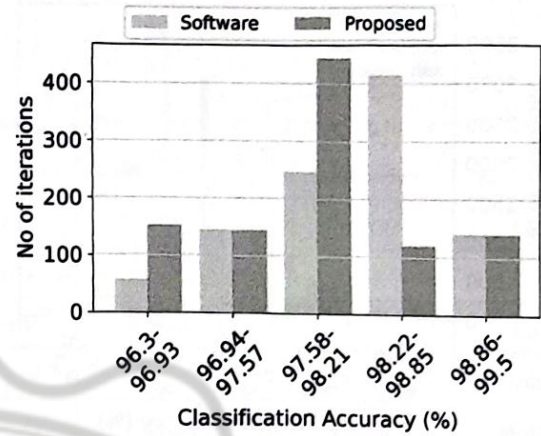


Fig. 17. Classification results of the LVQ (based on SFC) and the equivalent software model on the brain tumor dataset over 1000 iterations.

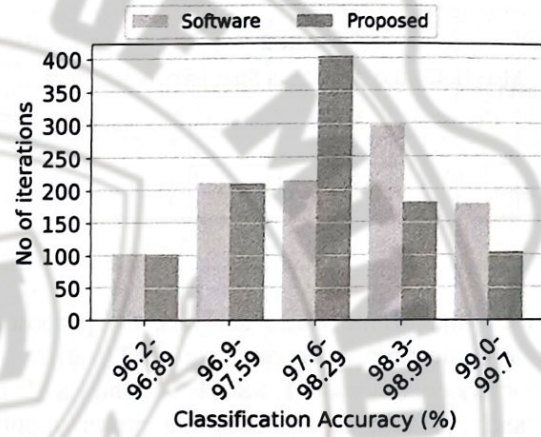


Fig. 18. Classification results of the LVQ (based on the Squarer circuit) and the equivalent software model on the brain tumor dataset over 1000 iterations.

compared to software. This is primarily because the circuits generate an approximation of the requested functions (DCCs) but are not ideal. However, the training of the parameters forms the foundation of their ideal model. However, in terms of minimum and mean classification performance, the Euclidean circuit-based classifiers show a slight drop-off in contrast to all the other two configurations. On the other hand, the Euclidean configuration, despite its minor deficiency when it comes to accuracy, can be worth mentioning power savings since it leads to the lowest power consumption. Finally, based on the 1000 iterations' testing, the architecture that achieves both competitive classification results and decreased power consumption is the one with the Sigmoid distance metrics as codebook vector representatives.

Apart from the 1000-iteration test, the designed analog circuits should be tested for their sensitivity in PVT variations. Thus, a Monte Carlo analysis (over process and mismatch variations) was implemented with $N = 5000$ distinct points (6σ). The overall results are summarized in Fig. 19, and their statistical attributes are presented in Table V for Euclidean, Sigmoid, and Squarer implementations, accordingly. All three implementations are proven to be robust, maintaining a worst case accuracy of above 97%. At the same time, the calculated variance in all three cases is below the 1% mark, thus validating the acceptable sensitivity characteristics of the proposed classifiers.

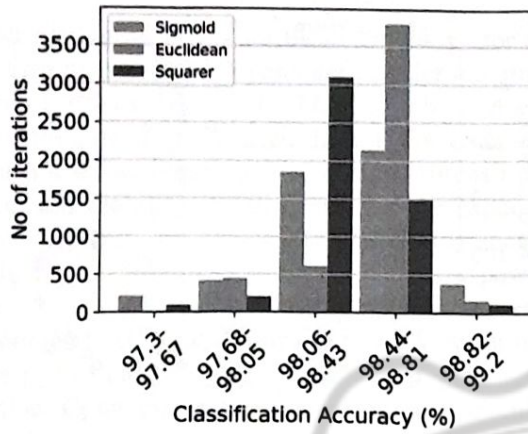


Fig. 19. Postlayout Monte Carlo simulation results of the LVQ (for each DCC circuit) on the brain tumor dataset.

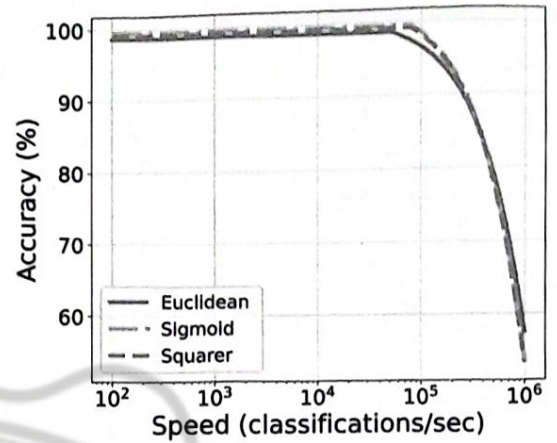


Fig. 20. Visual illustration of the tradeoff between the classifier's operation speed and classification accuracy.

TABLE V

MONTE CARLO ANALYSIS SIMULATION RESULTS

Classifier	Mean	Standard deviation
E-LVQ	98.55%	0.28%
Si-LVQ	98.38%	0.32%
Sq-LVQ	98.33%	0.29%

Apart from the Monte Carlo analysis, the proposed classifiers undergo testing to account for PVT variations. The selected corners encompass TT, SS, FF, SF, and FS (T: typical, S: slow, and F: fast). In addition, the power supply rails fluctuate within the range from $V_{DD} = -V_{SS} = 0.25$ V to $V_{DD} = -V_{SS} = 0.35$ V. Regarding temperature, the assessed spectrum spans from -25 °C to 125 °C. All three implementations exhibit resilience across corners, maintaining a minimum classification accuracy of 91.44%, 92.66%, and 93.08% for the Euclidean, Sigmoid, and Squarer circuits as DCC, respectively, under the worst case scenario. The most challenging corner scenario emerges with SS, -25 °C, and $V_{DD} = -V_{SS} = 0.25$ V, coupled with reduced software-based accuracy (worst case).

A key attribute in a classifier accelerator circuit is its processing speed while maintaining sufficient accuracy. Specifically, by computing the settling time of each individual block, the highest accuracy is attained when the classifier achieves a rate of 50k classifications per second for E-LVQ and 80k classifications per second for the other two. However, in scenarios where accuracy is less critical, this rate can be further augmented. It is important to acknowledge that increasing the classification speed may also exacerbate the classifier's power consumption. For example, E-LVQ exhibits lower power consumption compared to the other two implementations, but it also has a lower classification speed. Therefore, for the same speed value, it would have higher consumption but lower classification accuracy. In addition, because the Euclidean distance circuit operates in voltage mode, controlling currents through voltages leads to higher currents at the nodes compared to biasing directly with current. The tradeoff between classification speed and accuracy is depicted in Fig. 20 for each classifier. Furthermore, the area and power efficiency of the classifier's circuit enable the

parallel deployment of numerous identical classifiers, significantly enhancing the overall classification speed in practical settings.

VII. DISCUSSION AND COMPARISON

In the existing literature, it is evident that analog classifiers are typically designed as application-specific engines. This specialization presents a challenge when attempting to conduct a fair comparison across diverse implementations. Consequently, there exists an opportunity to tailor the design of these classifiers to cater to the same application, facilitating a comprehensive assessment of performance across various ML models and methodologies. Specifically, Table VI offers a performance summary of this work alongside related analog classifiers all within the brain tumor classification task.

All the summarized classifiers are implemented in a TSMC 90-nm CMOS process technology, with power supply rails selected based on the operating region and a tradeoff between higher accuracy and lower power consumption. All classifiers were trained using the required software, which relied on the mathematical models described in each implementation. Subsequently, they were all designed using the TSMC 90-nm CMOS process. At this stage, they underwent schematic-level verification (except our work, which is verified at the layout level too), and necessary enhancements were implemented to optimize classification accuracy and speed while prioritizing minimal power consumption. We followed the same design process as outlined in Section IV. In cases where the architecture operates in saturation, we applied the corresponding techniques specific to that operating region. The aforementioned process aimed to ensure a fair comparison, given that the implementations were carried out using different technologies and for distinct classification tasks. It includes a variety of analog classifiers, such as a radial basis function (RBF) [40], an RBF-neural network (NN) [41], [54], an artificial NN (ANN) [55], Bayes [42], a Gaussian mixture model (GMM) [14], support vector machine (SVM) [43], [44], a K -means [45], a support vector regression (SVR) [46], a support vector domain description (SVDD) [47], a self-organized map (SOM) [48], a long short-term memory (LSTM) [49], a multilayer perceptron (MLP) [50], a fuzzy [51], a threshold [52],

TABLE VI
ANALOG CLASSIFIERS' COMPARISON ON THE BRAIN TUMOR DATASET

	Classifier	Best accuracy (%)	Worst accuracy (%)	Mean accuracy (%)	Power consumption	Processing speed ($\frac{\text{classifications}}{s}$)	Energy (pJ) per classification	Architectures Complexity
This work	E-LVQ	98.70	96.00	97.34	547 nW	50K	10.94	Low
This work	Si-LVQ	99.20	96.30	97.83	643 nW	80K	8.04	Low
This work	Sq-LVQ	99.50	96.20	97.98	592 nW	80K	7.40	Low
[14]	GMM	95.90	89.70	92.51	2.08 μ W	120K	17.36	Medium
[40]	RBF	92.70	87.40	90.33	22.75 μ W	170K	133.82	Medium
[41]	RBF-NN	97.80	92.10	94.12	1.25 μ W	270K	4.63	Medium
[42]	Bayes	94.70	88.70	91.52	846 nW	120K	7.05	Low
[43]	SVM	96.40	92.70	94.66	892.2 μ W	870K	1030	High
[44]	SVM	96.10	91.20	93.77	63.1 μ W	140K	450.71	Medium
[45]	K-means	98.70	95.20	97.31	267.3 μ W	5M	53.46	High
[46]	SVR	97.20	96.20	96.57	87.2 μ W	870K	100.02	High
[47]	SVDD	96.90	95.90	96.21	55.7 μ W	530K	105.09	High
[48]	SOM	99.80	95.90	96.72	729 μ W	180K	4050	Medium
[49]	LSTM	100.00	98.10	99.57	63 mW	870M	72.41	Very High
[50]	MLP	99.80	96.80	97.58	1.02 mW	930K	1090	High
[51]	Fuzzy	98.10	91.20	95.39	1.01 μ W	4.55K	221.98	Medium
[52]	Threshold	97.20	92.40	95.87	428 nW	100K	4.28	Low
[53]	Centroid	98.20	91.30	95.28	3.72 μ W	170K	21.88	Medium
[54]	RBF NN	97.30	91.80	94.31	6.93 μ W	250K	27.72	Medium
[41]	RBF NN	96.90	91.50	94.12	8.53 μ W	310K	27.52	Medium
[55]	ANN	93.40	89.70	91.56	26.31 μ W	3M	8.77	Medium
[56]	SNN	99.30	96.80	97.88	28.32 μ W	350K	80.9	High
[57]	PM	96.10	92.30	94.41	89.71 μ W	180K	498.39	Medium

a cascaded-connected centroid [53], a spiking NN (SNN) [56], and a pattern-matching (PM) classifier [57].

The configurations outlined in Table VI rely on mathematical model approximations. Furthermore, the implementations referred to in [14], [40], [42], [43], [44], [48], [51], [52], and [53] incorporate Gaussian function (bump) circuits as their fundamental structural components. In these architectures, the power supply rails are set to $V_{DD} = -V_{SS} = 0.3$ V. For the remaining implementations, we selected power supply rails between $V_{DD} = -V_{SS} = 0.6$ V and $V_{DD} = -V_{SS} = 0.75$ V. These architectures operate in the saturation region, requiring a higher supply voltage. The foundational design principle of these endeavors centers on the utilization of multivariate Gaussian functions, resulting in the integration of cascaded circuits. At the circuit level, the bias current of each bump circuit becomes the output current of the preceding one. The primary limitation stems from the degradation of the current

from the input to the output of the multivariate Gaussian function circuit. In comparison to alternative studies, this work distinguishes itself by offering the ability to control weights for each individual feature, as opposed to adjusting the overall probability for the entire class. In addition, existing methodologies exhibit a constrained operating range for classifiers. If the chosen parameter from training is near the power supply edges, the output current decreases compared to a parameter at the center of the power supply. As a consequence, the output current of the Gaussian function circuit may decrease to a level below the operating current for subsequent circuits.

Regarding architectural complexity, there exists a spectrum of approaches, ranging from low to high complexity, with the specific ML model and the nature of the approximation influencing the level of complexity. In the evaluation of architectures, including E-LVQ, Si-LVQ, and Sq-LVQ emerges as the most effective in achieving high classification accuracy

and performance. This superiority is attributed to the quality of the Sq-LVQ architecture's approximation compared to other approaches. The proposed implementations surpass all other classifiers in Table VI concerning mean accuracy, except for MLP and LSTM algorithms, which excel in balancing model complexity and hardware-approximation efficiency. Notably, this heightened performance is achieved with the least energy consumption per classification compared to alternative approaches. While the threshold classifier achieves the lowest power consumption, it does so at the expense of accuracy and processing speed due to the simplicity of its model. Moreover, this work provides a tradeoff between power consumption, energy per classification, and classification accuracy, emphasizing the flexibility to sacrifice speed for power consumption in biomedical applications [58].

An additional point of interest is comparing LVQ with a high-complexity algorithm, as shown in Table VI. Specifically, LVQ is straightforward and interpretable, making it suitable for smaller datasets with clear class boundaries. It exhibits moderate scalability and performs well on datasets with a small number of features. In addition, it can achieve higher performance and greater power efficiency in all related classification tasks compared to simpler ML models that achieve good accuracy [14], [40], [42], [43], [44], [48], [51], [52], [53].

For the brain tumor classification task, we adapted several models to effectively handle the unique characteristics of our dataset (for comparison purposes). The LSTM model, typically designed for sequential data, was adapted by preprocessing feature vectors into temporal sequences. This approach allowed the model to capture sequential dependencies within the dataset, crucial for identifying subtle patterns indicative of different tumor types. Although LSTM models are typically used for temporal data, we adapted them for this task by considering sequences of feature vectors over multiple instances as pseudotime steps. This approach allowed the LSTM to capture underlying patterns within the dataset. *K*-means is primarily a clustering algorithm and is not directly suitable for classification without additional steps. It has high scalability but limited performance for classification without additional adaptations. For *K*-means, we first clustered the data into distinct groups. Each cluster was then assigned a class label based on the majority of the training samples within the cluster. For new samples, the nearest cluster centroid determined the predicted class.

SVDD is specialized for anomaly detection, may not perform well on traditional classification tasks, and has moderate scalability. SVR was originally designed for regression but can be adapted for classification using additional techniques. It has moderate scalability with variable performance depending on hyperparameter tuning and data characteristics. More specifically, it was adapted by training it to predict a continuous value that was then mapped to class labels. This mapping was done by establishing thresholds that corresponded to different classes, effectively converting the regression task into a classification task. SOM was used to create a topological map of the input data. During the classification phase, each input was mapped to the nearest node on the SOM grid, and

the class label was assigned based on the majority class of the training samples mapped to that node.

ANN is versatile and suitable for various classification tasks but requires tuning to prevent overfitting, and it has high scalability. SNN is appropriate for tasks requiring temporal processing and event-driven computation but may necessitate specific training methods. Also, it has high scalability. Furthermore, MLP is versatile and capable of handling various classification tasks with high scalability but may require careful tuning to avoid overfitting. The ANN model used in this study refers to a simple feedforward NN with a single hidden layer, while the MLP model incorporates multiple hidden layers and more sophisticated training techniques, such as dropout and batch normalization, to enhance performance and prevent overfitting.

In addition, we considered feature dimensionality and model complexity in our implementations. For instance, models with higher complexity and more parameters, such as deep MLPs, were carefully tuned to balance accuracy and energy efficiency. We observed that increasing the feature dimensionality generally improved classification accuracy but also led to higher energy consumption. Therefore, feature selection techniques were employed to identify the most informative features, reducing dimensionality while maintaining high accuracy. The choice of parameters, such as feature dimensionality and model complexity, directly impacts the energy efficiency per classification for different models. By optimizing these parameters, we were able to achieve a balance between high performance and low power consumption across the related models.

VIII. CONCLUSION

This study presented a design methodology focused on analog integrated architecture for the LVQ algorithm, targeting low-power applications and achieving high accuracy (more than 96.00%). The high-level architecture consists of DCCs, CMs, and a CC. Three primary approaches were established, distinguished by the circuit used for DCC. All implementations were power-efficient (less than 650 nW) and low supply voltage (only 0.6 V). Also, they are robust under PVT variations both over Corners and Monte Carlo simulations. Each specific approach was tested in a brain tumor classification task and compared with a software-based implementation and related analog classifiers. The designs were developed and simulated in a 90-nm CMOS process using the Cadence IC Suite.

REFERENCES

- [1] S. Rasool, A. Husnain, A. Saeed, A. Y. Gill, and H. K. Hussain, "Harnessing predictive power: Exploring the crucial role of machine learning in early disease detection," *JURIHUM, Jurnal Inovasi Dan Humaniora*, vol. 1, no. 2, pp. 302–315, 2023.
- [2] K. Dzobo, S. Adotey, N. E. Thomford, and W. Dzobo, "Integrating artificial and human intelligence: A partnership for responsible innovation in biomedical engineering and medicine," *OMICS, A J. Integrative Biol.*, vol. 24, no. 5, pp. 247–263, May 2020.
- [3] M. S. Ibrahim and S. Saber, "Machine learning and predictive analytics: Advancing disease prevention in healthcare," *J. Contemp. Healthcare Anal.*, vol. 7, no. 1, pp. 53–71, 2023.

- [4] P. Manickam et al., "Artificial intelligence (AI) and Internet of Medical Things (IoMT) assisted biomedical systems for intelligent healthcare," *Biosensors*, vol. 12, no. 8, p. 562, Jul. 2022.
- [5] A. Amar, A. Kouki, and H. Cao, "Power approaches for implantable medical devices," *Sensors*, vol. 15, no. 11, pp. 28889–28914, Nov. 2015.
- [6] E. Locorotondo, V. Cultrera, L. Pugi, L. Berzi, M. Pierini, and G. Lutzemberger, "Development of a battery real-time state of health diagnosis based on fast impedance measurements," *J. Energy Storage*, vol. 38, Jun. 2021, Art. no. 102566.
- [7] B. C. S. Loh and P. H. H. Then, "Deep learning for cardiac computer-aided diagnosis: Benefits, issues & solutions," *mHealth*, vol. 3, pp. 1–45, Oct. 2017.
- [8] F. Simon and G. Giovannetti, *Managing Biotechnology: From Science to Market in the Digital Age*. Hoboken, NJ, USA: Wiley, 2017.
- [9] W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog computing," *Proc. IEEE*, vol. 107, no. 1, pp. 108–122, Jan. 2019.
- [10] V. Alimisis, N. P. Eleftheriou, A. Kamperi, G. Gennis, C. Dimas, and P. P. Sotiriadis, "General methodology for the design of bell-shaped analog-hardware classifiers," *Electronics*, vol. 12, no. 20, p. 4211, Oct. 2023.
- [11] M. A. Hannan, S. M. Abbas, S. A. Samad, and A. Hussain, "Modulation techniques for biomedical implanted devices and their challenges," *Sensors*, vol. 12, no. 1, pp. 297–319, Dec. 2011.
- [12] M.-P. Hosseini, A. Hosseini, and K. Ahi, "A review on machine learning for EEG signal processing in bioengineering," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 204–218, 2021.
- [13] R. Akerkar and P. S. Sajja, "Bio-inspired computing: Constituents and challenges," *Int. J. Bio-Inspired Comput.*, vol. 1, no. 3, p. 135, 2009.
- [14] V. Alimisis, G. Gennis, K. Touloupas, C. Dimas, M. Gourdouparis, and P. P. Sotiriadis, "Gaussian mixture model classifier analog integrated low-power implementation with applications in fault management detection," *Microelectron. J.*, vol. 126, Aug. 2022, Art. no. 105510.
- [15] K. Arshak, A. Arshak, E. Jafer, D. Waldern, and J. Harris, "Low-power wireless smart data acquisition system for monitoring pressure in medical application," *Microelectron. Int.*, vol. 25, no. 1, pp. 3–14, 2007.
- [16] F. Hu, S. Lakdawala, Q. Hao, and M. Qiu, "Low-power, intelligent sensor hardware interface for medical data preprocessing," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 4, pp. 656–663, Jul. 2009.
- [17] E. Spano, S. Di Pascoli, and G. Iannaccone, "Low-power wearable ECG monitoring system for multiple-patient remote monitoring," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5452–5462, Jul. 2016.
- [18] S. Hu, H. Wei, Y. Chen, and J. Tan, "A real-time cardiac arrhythmia classification system with wearable sensor networks," *Sensors*, vol. 12, no. 9, pp. 12844–12869, Sep. 2012.
- [19] M. Yang, C.-H. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "A $1\mu\text{W}$ voice activity detector using analog feature extraction and digital deep neural network," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2018, pp. 346–348.
- [20] L. Yan et al., "24.4 A 680nA fully integrated implantable ECG-acquisition IC with analog feature extraction," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, Feb. 2014, pp. 418–419.
- [21] M. Yang et al., "Nanowatt acoustic inference sensing exploiting non-linear analog feature extraction," *IEEE J. Solid-State Circuits*, vol. 56, no. 10, pp. 3123–3133, Oct. 2021.
- [22] M. Hock, A. Hartel, J. Schemmel, and K. Meier, "An analog dynamic memory array for neuromorphic hardware," in *Proc. Eur. Conf. Circuit Theory Des.*, 2013, pp. 1–4.
- [23] C. Bachmann, M. Ashouei, V. Pop, M. Vidojkovic, H. D. Groot, and B. Gyselinckx, "Low-power wireless sensor nodes for ubiquitous long-term biomedical signal monitoring," *IEEE Commun. Mag.*, vol. 50, no. 1, pp. 20–27, Jan. 2012.
- [24] *Brain Tumor*. Accessed: Nov. 8, 2023. [Online]. Available: <https://www.kaggle.com/dsv/1370629>
- [25] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [26] *Learning Vector Quantization for Pattern Recognition*, document TKK-F-A602, 1990.
- [27] D. Mukherjee and S. K. Mitra, "Successive refinement lattice vector quantization," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1337–1348, Dec. 2002.
- [28] A. Gopalan and A. H. Titus, "A new wide range Euclidean distance circuit for neural network hardware implementations," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1176–1186, Sep. 2003.
- [29] B. Gilbert, "Translinear circuits: An historical overview," *Anal. Integr. Circuits Signal Process.*, vol. 9, no. 2, pp. 95–118, Mar. 1996.
- [30] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of $O(n)$ complexity," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 1988, pp. 1–26.
- [31] K. M. Al-Tamimi and M. A. Al-Absi, "An ultra low power high accuracy current-mode CMOS squaring circuit," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 2, 2012, pp. 1–23.
- [32] G. N. Patel and S. P. DeWeerth, "An analog VLSI loser-take-all circuit," in *Proc. Int. Symp. Circuits Syst.*, vol. 2, 1995, pp. 850–853.
- [33] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-Threshold Design for Ultra Low-Power Systems*, vol. 95. Cham, Switzerland: Springer, 2006.
- [34] K. Liu and D. Indiveri, *Analog VLSI: Circuits and Principles*. MIT Press, 2002.
- [35] A. Tajalli and Y. Leblebici, *Extreme Low-Power Mixed Signal IC Design: Subthreshold Source-coupled Circuits*. Cham, Switzerland: Springer, 2010.
- [36] C. Mead, "Analog VLSI and neural systems," *NASA STI/Recon Technical Report A*, vol. 90, Jun. 1989, Art. no. 16574.
- [37] L. Kapoor and S. Thakur, "A survey on brain tumor detection using image processing techniques," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2017, pp. 582–585.
- [38] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.
- [39] R. Li and H. Fariborzi, "Ultra-low power data converters with BEOL NEM relays," in *Proc. IEEE 61st Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2018, pp. 627–630.
- [40] S.-Y. Peng, P. E. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2148–2158, Oct. 2007.
- [41] A. R. Mohamed, L. Qi, Y. Li, and G. Wang, "A generic nano-watt power fully tunable 1-D Gaussian kernel circuit for artificial neural network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 9, pp. 1529–1533, Sep. 2020.
- [42] V. Alimisis, G. Gennis, C. Dimas, and P. P. Sotiriadis, "An analog Bayesian classifier implementation, for thyroid disease detection, based on a low-power, current-mode Gaussian function circuit," in *Proc. Int. Conf. Microelectron. (ICM)*, Dec. 2021, pp. 153–156.
- [43] 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.
- [44] V. Alimisis, G. Gennis, M. Gourdouparis, C. Dimas, and P. P. Sotiriadis, "A low-power analog integrated implementation of the support vector machine algorithm with on-chip learning tested on a bearing fault application," *Sensors*, vol. 23, no. 8, p. 3978, Apr. 2023.
- [45] R. Zhang and T. Shibata, "An analog on-line-learning k-means processor employing fully parallel self-converging circuitry," *Anal. Integr. Circuits Signal Process.*, vol. 75, pp. 267–277, May 2013.
- [46] R. Zhang, N. Uetake, T. Nakada, and Y. Nakashima, "Design of programmable analog calculation unit by implementing support vector regression for approximate computing," *IEEE Micro*, vol. 38, no. 6, pp. 73–82, Nov. 2018.
- [47] R. Zhang and T. Shibata, "A VLSI hardware implementation study of SVDD algorithm using analog Gaussian-cell array for on-chip learning," in *Proc. 13th Int. Workshop Cellular Nanosc. Netw. Appl.*, Aug. 2012, pp. 1–6.
- [48] F. Li, C.-H. Chang, and L. Siek, "A compact current mode neuron circuit with Gaussian taper learning capability," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 2129–2132.
- [49] Z. Zhao, A. Srivastava, L. Peng, and Q. Chen, "Long short-term memory network design for analog computing," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 1, pp. 1–27, Jan. 2019.
- [50] K. Lee, J. Park, and H.-J. Yoo, "A low-power, mixed-mode neural network classifier for robust scene classification," *J. Semicond. Technol. Sci.*, vol. 19, no. 1, pp. 129–136, Feb. 2019.
- [51] E. Georgakilas, V. Alimisis, G. Gennis, C. Aletraris, C. Dimas, and P. P. Sotiriadis, "An ultra-low power fully-programmable analog general purpose type-2 fuzzy inference system," *Int. J. Electron. Commun.*, vol. 170, Oct. 2023, Art. no. 154824.

- [52] V. Alimisis, G. Gennis, E. Tsouvalas, C. Dimas, and P. P. Sotiriadis, "An analog, low-power threshold classifier tested on a bank note authentication dataset," in *Proc. Int. Conf. Microelectron. (ICM)*, Dec. 2022, pp. 66–69.
- [53] V. Alimisis, V. Mouzakis, G. Gennis, E. Tsouvalas, and P. P. Sotiriadis, "An analog nearest class with multiple centroids classifier implementation, for depth of anesthesia monitoring," in *Proc. Int. Conf. Smart Syst. Power Manage.*, Nov. 2022, pp. 176–181.
- [54] A. Dorzhigulov and A. P. James, "Generalized bell-shaped membership function generation circuit for memristive neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
- [55] S. T. Chandrasekaran, R. Hua, I. Banerjee, and A. Sanyal, "A fully-integrated analog machine learning classifier for breast cancer classification," *Electronics*, vol. 9, no. 3, p. 515, Mar. 2020.
- [56] E. Donati, M. Payvand, N. Risi, R. Krause, and G. Indiveri, "Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 795–803, Oct. 2019.
- [57] T. Yamasaki and T. Shibata, "Analog soft-pattern-matching classifier using floating-gate mos technology," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1257–1265, Sep. 2003.
- [58] H.-T. Wu, "Current state of nonlinear-type time–frequency analysis and applications to high-frequency biomedical signals," *Current Opinion Syst. Biol.*, vol. 23, pp. 8–21, Jul. 2020.



Vassilis Alimisis (Graduate Student Member, IEEE) received the B.Sc. degree in physics (top 1%) and the M.Sc. degree in electronics and communications from the University of Patras, Patras, Greece, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the National Technical University of Athens (NTUA), Athens, Greece, under the supervision of Prof. Paul P. Sotiriadis. His Ph.D. thesis and research are supported and financed by the E.L.K.E. NTUA Scholarships.

He is a Teaching Assistant in undergraduate and graduate courses and supervises diploma thesis. He has authored or co-authored several conference papers and journal articles. His research interests include analog microelectronic circuits, low-power electronics, analog computing, and integrated circuit architectures with applications in artificial intelligence and machine learning.

Mr. Alimisis received the Best Paper Award in the IEEE International Conference on Microelectronics in 2020, the Best Paper Award in the IEEE International Conference on Microelectronics in 2021, the Best Paper Award (Third Place) in the IEEE International Conference on Microelectronics in 2023, the Best Paper Award in IEEE Symposium on Integrated Circuits and Systems Design (SBCCI) in 2021, and the Best Paper Award in the First International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023. He regularly reviews for many IEEE TRANSACTIONS and conferences and serves on proposal review panels.



Emmanouil Anastasios Serlis (Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2023, where he is currently working toward the Ph.D. degree under the supervision of Prof. Paul P. Sotiriadis.

He has co-authored several conference papers and journal articles. His research interests include analog microelectronic circuits, ultralow-power electronics, analog computing, and integrated circuit architectures with applications in artificial intelligence, machine learning, and deep neural networks.



Andreas Papathanasiou (Student Member, IEEE) is currently working toward the Diploma degree from the National Technical University of Athens (NTUA), Athens, Greece, under the supervision of Prof. Paul P. Sotiriadis.

He is a Senior Graduate Student with the Department of Electrical and Computer Engineering, NTUA. He has co-authored a journal article. His research interests include analog microelectronic circuits, ultralow-power electronics, analog computing, and integrated circuit architectures with applications



Nikolaos P. Eleftheriou (Member, IEEE) is currently pursuing the Diploma degree under the supervision of Prof. Paul P. Sotiriadis.

He is a Senior Graduate Student with the Department of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece. He has co-authored several conference papers and journal articles. His research interests include microelectronic circuit design, analog circuits, and analog hardware computing techniques with applications in fuzzy systems, artificial intelligence, and machine learning.

Mr. Eleftheriou was a recipient of the Panagiotis Triantafyllidis Scholarship for undergraduate studies. He received the Best Paper Award (Third Place) in the IEEE International Conference on Microelectronics (ICM) in 2023 and the Best Paper Award in the First International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023.



Paul P. Sotiriadis (Fellow, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1994, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1996, and the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2002.

In 2002, he joined a Faculty Member with the Electrical and Computer Engineering Department, Johns Hopkins University, Baltimore, MD, USA. In 2012, he joined a Faculty Member with the Electrical and Computer Engineering Department, NTUA, where he is currently a Professor of electrical and computer engineering and the Director of the Electronics Laboratory. He is a Governing Board Member of the Hellenic (National) Space Center of Greece. He runs a team of 25 researchers. He has authored or co-authored more than 200 research publications, most of them in IEEE journals and conferences, holds one patent, and has contributed several chapters to technical books. He has been on the list of the top 2% most influential researchers in the world in 2020, 2022, and 2023. His research interests include the design, optimization, and mathematical modeling of analog, mixed-signal, and RF integrated and discrete circuits, sensor and instrumentation architectures with an emphasis on biomedical instrumentation, advanced RF frequency synthesis, the application of machine learning and general AI in the operation, and the design of electronic circuits.

Prof. Sotiriadis received several awards, including the prestigious Guillemin-Cauer Award from the IEEE Circuits and Systems Society in 2012, the Best Paper Award in the First International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023, the Best Paper Award (Third Place) in the IEEE International Conference on Microelectronics (ICM) in 2023, the Best Paper Award in ICM in 2021, the Best Paper Award in the IEEE Symposium on Integrated Circuits and Systems Design (SBCCI) in 2021, the Best Paper Award in the ICM in 2020, the Best Paper Award in the IEEE International Conference on Modern Circuits and Systems Technologies in 2019, the Best Paper Award in the IEEE International Frequency Control Symposium in 2012, the Best Paper Award in the IEEE International Symposium on Circuits and Systems in 2007, and the IEEE Circuits and Systems Society (CASS) Outstanding Technical Committee Recognition in 2022. He is an Associate Editor of the IEEE SENSORS JOURNAL and IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2016 to 2020 and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS from 2005 to 2010. He has been a member of technical committees of many conferences. He regularly reviews for many IEEE TRANSACTIONS and conferences and serves on proposal review panels.

in artificial intelligence and machine learning.