Programmable Olfactory Computing

Nathaniel Bleier Abigail Wezelis Lav Varshney Rakesh Kumar
nbleier3@illinois.edu
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

ABSTRACT

While smell is arguably the most visceral of senses, olfactory computing has been barely explored in the mainstream. We argue that this is a good time to explore olfactory computing since a) a large number of driver applications are emerging, b) odor sensors are now dramatically better, and c) non-traditional form factors such as sensor, wearable, and xR devices that would be required to support olfactory computing are already getting widespread acceptance. Through a comprehensive review of literature, we identify the key algorithms needed to support a wide variety of olfactory computing tasks. We profiled these algorithms on existing hardware and identified several characteristics, including the preponderance of fixed-point computation, and linear operations, and real arithmetic; a variety of data memory requirements; and opportunities for data-level parallelism. We propose Ahromaa, a heterogeneous architecture for olfactory computing targeting extremely power and energy constrained olfactory computing workloads and evaluate it against baseline architectures of an MCU, a state-of-art CGRA, and an MCU with packed SIMD. Across our algorithms, Ahromaa's operating modes outperform the baseline architectures by 1.36, 1.22, and 1.1× in energy efficiency when operating at MEOP. We also show how careful design of data memory organization can lead to significant energy savings in olfactory computing, due to the limited amount of data memory many olfactory computing kernels require. These improvements to the data memory organization lead to additional 4.21, 4.37, and 2.85× improvements in energy efficiency on average.

ACM Reference Format:
Nathaniel Bleier Abigail Wezelis Lav Varshney Rakesh Kumar. 2023. Programmable Olfactory Computing. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA '23), June 17–21, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3579371.3589061

1 INTRODUCTION

Arguably the biggest development in computing in the last few decades has been the advent and proliferation of sensor driven computing [72]. Use of image, health, audio, and other sensors have resulted in novel computing platforms such as wearables [8, 10, 101], head mounted displays [6, 62], and smartphones [104], and novel computing applications such as health and wellness tracking [15],

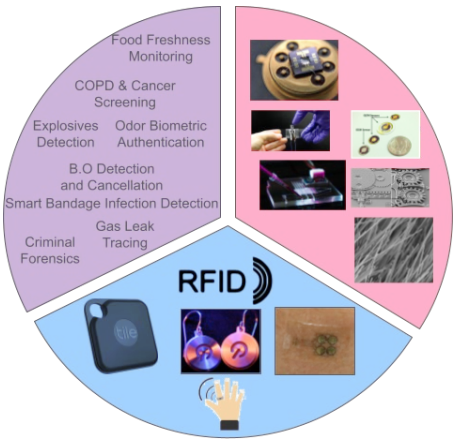


Figure 1: An overview of olfactory computing applications, form-factors, and sensors & actuators.

digital navigation [120], and extended reality [60]. Indeed, each new class of sensors can potentially enable new applications and generate new computational requirements.

One class of sensor data processing that has been barely explored in the mainstream is *olfactory computing*. Smell has always received scant recognition [100], and has historically been considered an inferior sense [100], in spite of arguably being the most visceral of senses [109]. Biological understanding of odor has also been relatively recent [11]. Previous attempts at olfactory computing, in the form of e-noses [68], for example, have seen limited success [96, 105], discouraging further research and adoption; some attempts have indeed been publicly ridiculed [105]. Finally, odor sensors that olfactory computing would rely on have historically been highly inaccurate, power hungry, and slow [126]; pumps and cartridges required for any odor synthesis [3, 9, 12, 16, 117] application have been similarly big, slow, and power hungry [4, 83, 107].

In spite of the checkered history, we believe that this is a good time to revisit olfactory computing. A large number of driver applications are emerging that would require or benefit from olfactory computing. Low cost healthcare (e.g., smart bandages for infection detection [37]) and personal hygiene applications (body odor detection [63–65, 127]), for example, could benefit greatly when augmented with odor sensing and synthesis. Second, there have been dramatic recent advances in odor sensing technology, with some modern odor sensors demonstrating high accuracy, miniaturized to the micron scale, and consuming no more than low tens of μW of power (Fig. 2). There have been similar advances in NEMS/MEMS-based micropumps and cartridges [117]. Third, widespread acceptance of sensor, wearable, and xR devices over the last decade, including recent interest in odor-based wearable [9, 12, 16, 17, 20, 95, 117] and xR [109] devices, suggests a much

ISCA '23, June 17–21, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0095-8/23/06…$15.00
https://doi.org/10.1145/3579371.3589061

easier path of adoption for olfactory computing-based applications than ever before - overlaid on these devices. This vision of olfactory computing is depicted in Fig. 1 which shows example odor applications, form factors for olfactory computing-enabled devices, and odor sensors and actuators.

Of course, to build olfactory computing systems, we must understand and analyze characteristics of odor-based applications. In this work, through a survey of research literature and recent practice, and conversations with olfaction experts, we first identify a set of computation tasks that may be common across a variety of odor-based applications (Sec. 2). We then identify, through a survey of papers discussing implementations of these tasks, a set of algorithms that often underpin these tasks (Sec. 2). We then understand through profiling-based measurements the computational characteristics of these algorithms (Sec. 3). An understanding of these characteristics can be used to build and optimize olfactory computing systems for different system-level constraints.

As a case study, we consider the architecture of a programmable olfactory computing hardware platform in the context of sensor and wearable applications with extreme power and energy constraints. Programmability may be essential in this context to lower costs and to maintain generality in a domain where applications are likely to evolve.

We observe (Sec. 3) that a programmable architecture for olfactory computing needs to support efficiently a) fixed-point multiply-accumulates and activations for various quantization levels, which are needed for many algorithms supporting olfactory tasks – many olfactory workloads are ANN-based or otherwise use linear algebraic kernels, b) general purpose and dataflow computation to support a variety of algorithms such as principal component analysis, K-Means, orthogonal matching pursuit, and particle filtering which are common across tasks, including support for computation kernels admitting data level parallelism, such as matrix decompositions, matrix multiplication, and limited non-linear scalar and serial operations such as exponentiation, square root, and division, and c) a configurable memory system to support "small data" algorithms, which, in the context of olfactory computing, include linear discriminant analysis, k-nearest neighbors, and support vector machines. We present Ahromaa - A **h**eterogeneous **r**econfigurable **o**dor **m**onitoring and **a**nalysis **a**rchitecture (Sec. 4) - that consists of a general purpose RISC-V core, a packed SIMD unit, and a CGRA supporting 4, 8, 16, and 32 bit fixed point arithmetic as a programmable platform for olfactory computing. Ahromaa is used in conjunction with a variety of memory configurations to deliver high energy efficiency on olfactory computing workloads (Sec. 6).

This paper makes the following contributions:

- A comprehensive review of the olfactory computing literature, identifying key algorithms needed to support a wide variety of olfactory computing tasks.
- Analysis of these algorithms within the context of olfactory computing (e.g., input and model sizes, performance and system constraints, and sensor array designs are all relevant in the analysis of common algorithms such as artificial neural networks in the olfactory context).
- Design and evaluation of Ahromaa — a heterogeneous architecture for olfactory computing — the first sensor processor designed for the unique constraints and limitations of olfactory computing.
- Description and evaluation of optimizations to Ahromaa's data memory organization which provide significant improvement to Ahromaa's energy efficiency at its minimal energy operating points.

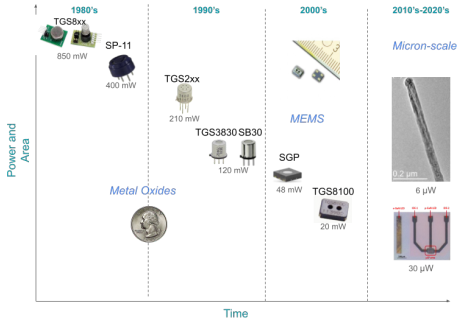


Figure 2: Odor and chemical sensors continue to improve in power, area, and response times.

Table 1: Examples of olfactory computing applications and their constraints.

Application	Constraints
Food Freshness Monitoring	≤ mW, lifetime of hours - months
Food Spoilage Detection	≤ mW, lifetime of hours - months
Personal Hygiene Monitoring	low mW, lifetime of ≤ day
Wearable Body Odor Detection	low mW, lifetime of days
Odor Biometric Authentication	lifetime of months-years
Odor Biometric Forensics	field deployable & single use
Smart-Bandage infection detection	< mW, lifetime of hours - days
Air and water quality monitoring	< mW, lifetime of months - years
Dangerous compound detection	lifetime of days-years
Explosive detection	lifetime of days-years
Gas Leak Tracing	requires spatial dispersed sensors, lifetime of minutes to hours
Odor Cancellation	requires odor synthesis
Bespoke clothing deodorization	requires odor synthesis
Olfactory enabled xR	requires odor synthesis
Food & Scent recommendation	
COPD & Lung Cancer Screening	

2 OLFACTORY TASKS AND ALGORITHMS

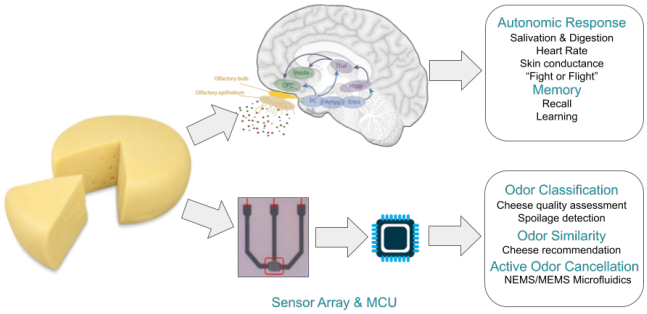


Figure 3: Human vs computerized olfactory systems.

The human olfactory system transduces chemicals into electrical signals in the olfactory epithelium. Signals are then broadcast to a number of different parts of the brain, which enable autonomic nervous system responses, and can conjure memories and enable

learning. This is mimicked in an odor sensor processor, where chemical sensor arrays transduce chemicals into odor signals which are sent to a compute element to execute odor related tasks (Fig. 3).

To build olfactory computing systems, we must understand and analyze the computational tasks underpinning odor-based applications. To identify a set of computation tasks that may be common across a variety of odor-based applications, we surveyed 25 papers published in last 10 years on odor-based applications in Table 1 - at least one paper on each application was found. We identified the computational task(s) each paper was focused on. We also surveyed 10 odor-based electronic products in last five years and identified the corresponding computational task(s). Finally, we interviewed three olfaction experts and asked them for a list of computational tasks. We then created a list of tasks that appeared in at least two of the above three lists.

2.1 Tasks

The tasks in our finalized list are odor localization, odor classification, odor authentication, odor similarity, active odor cancellation, odor pleasantness estimation, and odor demixing.

Odor classification, as the name suggests, is the task associated with identifying the source of an odor. Odor classification uses machine learning and statistical techniques to map chemical sensor readings to source odorants [59, 67]. Odor classification has many commercial, industrial, agricultural, medical, and security applications [88, 131] [28, 36, 114, 116], [23, 38, 74, 125], [81] [22, 42, 47, 118], [26, 80, 113]. Odor classification is performed with a variety of machine learning and statistical analysis techniques such as principal component analysis (PCA), support vector machines (SVM), artificial neural networks (ANN), and K-Means cluster analysis (KMeans).

Biometric authentication using odor, or, odor authentication, is the olfactory equivalent of fingerprint, facial, iris, or voice authentication techniques which are currently in vogue on mobile devices [110]. Individual humans have an identifiable scent due to genetic and environmental factors [94], and many studies have shown successful discrimination of individuals' body odor using chemical sensor data [63, 64, 127]. Odor authentication consists of first extracting features from chemical sensor data, often using parametric machine learning techniques, and then comparing those features to a user dictionary. Algorithms in odor authentication include PCA, SVM, ANN, K-Means, and KNN.

In the odor similarity task, the goal is to estimate the odor similarity of two different chemical mixtures. Traditionally, this has been done by first identifying the chemical structure of the mixtures' constituent chemicals via gas chromatography-mass spectrometry (however, it can also be done with sensor data). These chemical structures are then mapped to fixed length vectors from an inner product space, and then the mixtures are represented by weighted sums of the constituent chemical vectors. Finally, the similarity of the two chemicals is scored based on the angle between the two vectors (AngleDist) [106].

Active odor cancellation is the olfactory equivalent of active noise cancellation. Active odor cancellation attempts to 'block out' one or more malodors in an environment by *adding* odors which cancel the malodor, rendering it as olfactory white noise [119]. The algorithm which determines how much of each odor to add to the environment consists of solving a quadratic program which, in many cases, is convex. As such, gradient descent based optimizers may be used to find local, and often, global solutions to the optimization problem.

The pleasantness of an odor is highly dependent on the physical properties of the odorant molecular structure [19]. As such, odor pleasantness estimation is assigning a value (either numeric or categorical) to a chemical corresponding to its predicted pleasantness. This has been achieved using PCA and SVM [77, 78, 102].

Odorants exist in mixtures, including time-evolving mixtures, yet often only one chemical odorant is of interest. Odor demixing is the task which isolates the signal corresponding to the chemical of interest. Compressive sensing, namely, orthogonal matching pursuit (OMP), has been shown to be useful in the olfactory domain since individual odors are sparse within the large space in which an odor can be represented.

Odor localization is the task of identifying the source location of an odorant 'plume' within an environment. Biology inspired methods, as well as probabilistic methods such as particle filtering are commonly used in conjunction with a model of plume dynamics [121]. While odor localization is typically performed by either a mobile field robot [32], or by a distributed sensor network [56], as odor localization benefits from taking chemical readings in multiple locations throughout the environment, we also consider future applications where a wearable aids a moving human or an animal in identifying an odor source (gas leak or fire source, for example).

2.2 Algorithms

For identifying key olfactory kernels and their characteristics, we use systematic-style review [18], a scholarly synthesis methodology common in fields such as health sciences to reduce bias. A key step in systematic-style review is to use keywords to index into a research database to find the relevant studies for further analysis. We use keywords for different olfactory tasks to index into Google Scholar to find the most well-cited papers about those tasks - these papers are then analyzed to identify the key algorithmic kernels used in those tasks and their typical input sizes and rates, implementations, and performance requirements.

- Select a keyword for each given olfactory computing task.
- Collect the top 50 most relevant papers returned by Google Scholar for the selected keyword.
- Sort the papers by the number of citations per year.
- Take up to the top ten relevant papers from this list.
- Any algorithm that is used two or more times in these papers is added to the list of algorithms for the corresponding task.
- Algorithmic metaparameters (e.g., neural network architectures, dimensionality, etc.) are taken by using the most heavyweight version of the algorithms found in the search (see Table 2). In one case, metaparameters are determined by memory availability on the baseline architecture (i.e., the number of particles used in particle filtering).

Fig. 4 is a graphical depiction of the relationship between olfactory computing applications (hexagons), the tasks (ellipses) from which the applications are composed, and the computational kernels (rectangles) which can be used to implement the task. Some

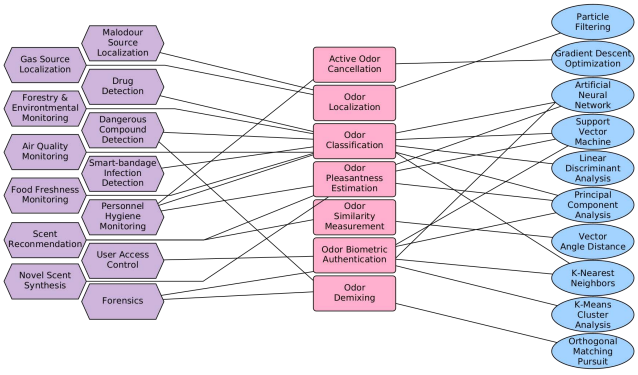


Figure 4: Graphical depiction of relationship between applications (hexagons) which are composed from tasks (rectangles). Tasks are implemented using a computational kernel (ellipses). Some tasks can be supported by multiple kernels.

tasks may be implemented using one of several kernels, such as odor classification which can be supported by several machine learning and statistical analysis techniques. Below we describe the nuances of implementation of the identified computational kernels in context of olfactory computing.

Particle filtering (PF) is an iterative statistical, Monte Carlo method used in odor source localization. The particle filter consists of a collection of 'particles'. The particles are assigned a location in the environment and a weight. Initially, the particle locations are chosen at random, and the weights are of each particle are equal and sum to one. Each iteration consists of 1) sampling the odor sensors, 2) updating the particle weights, 3) removing particles with low weights, 4) splitting particles with high weights, and 5) replacing some particles at random. If a particle gets sufficiently high weight, the particle filter terminates, and the location of the particle is determined to be the location of the odorant source. The key idea is that the weights are updated based on the likelihood that the source is at the position of the particle given the sequence of sensor readings and a dynamical model of the odorant plume.

Principal component analysis (PCA) is an unsupervised statistical analysis algorithm, commonly used for dimensionality reduction and feature extraction for sensor-based applications. For olfactory computing applications, due to the slow rate at which odorant mixtures change, the number of data points, d, in the analysis is limited to the low hundreds, while the number of sensors in a sensor array is often less than ten. Thus PCA requires normalizing only a small number of distributions with a small number of samples, and performing eigen decomposition on a small correlation matrix. Similarly, in the odor context, finding the principal components of a sensed input (PCA-Inf) requires only multiplying the input by a small $n \times n$ matrix.

Linear discriminant analysis (LDA) is a supervised statistical analysis algorithm which can be performed *offline*. For field olfactory computing applications, LDA consists of applying linear transformations to sensor data, and then assigning a class based on the linear transformation. As such, LDA is extremely light weight, with time and space complexity quadratic and linear, respectively, in the number of sensors in the sensor array.

The support vector machine (SVM) works by finding a hyperplane which separates the classes of data. Thus, determining if a sample belongs to class 0 or class 1 is as straightforward as determining which side of the hyperplane the sample lies by performing a linear transformation on the sample. In olfactory computing, the input sample is a vector of sensed values from the chemical sensor array. Thus, the complexity of running the support vector machine is the same as computing the principal components or linear discriminants of the input. Multiclass classifiers are built out of multiple SVMs with a voting mechanism.

Artificial neural networks (ANN) have been shown to be effective models for odor classification and odor biometrics. The most prominent type of ANN seen in olfactory computing is the multilayer perceptron (MLP), in which each layer is fully connected. The input to the MLP is typically the sensed data directly, with no preprocessing steps involved (consider, for example, that there is no need for preprocessing to help the model learn high-frequency content, as there is in image and audio processing, since odor signals typically lack high frequency content). The MLPs used in olfactory computing tend to be just barely deep - often with only a single hidden layer - and with very few neurons (often < 50). Straying from the trend of MLPs, [128] proposes a convolutional neural network (CNN) with two convolution layers and a fully connected layer which processes a time-series of 250 sensor array inputs. The CNN is used to estimate odor pleasantness.

K-Means Cluster Analysis (KMeans) is an unsupervised technique which partitions a collection of points in a metric space into k 'clusters', based on the proximities of the points in the dataset to one another. Termination of this algorithm is known to be NP-Hard [14]. In practice, for odor processing, KMeans often terminates quickly [130, 135].

Orthogonal matching pursuit (OMP) is an algorithm for compressive sensing, and is a variant of matching pursuit. OMP is an iterative algorithm which enables correctly recovering an odor signal from sparse sampling with high probability. We model our implementation of OMP from the example given in in [40]. The size of the input to OMP corresponds to the number of sensors used in the sensor array.

K-Nearest Neighbors (KNN) is a supervised algorithm used in classification and biometrics tasks. The complexity of this algorithm, regardless of its exact implementation, scales with k, the dimension of the space, and the number of points in the database. In olfactory computing applications, the dimensionality is typically the number of sensors in the sensor array, and $k \leq 3$ [44]. The number of points in the database varies, but, in biometric contexts, is typically limited by the number of authorized users enrolled (which in the case of wearable electronics, is one).

Fig. 5 shows the fraction of time spent in an olfactory computing task for different olfactory computing applications[1] This fraction is significant — no less than 72.8% (Scent Recommendation) and up to 98% (Personal Hygiene Monitoring). An architecture optimized for the tasks will, therefore, also perform well on the applications.

[1]Applications were implemented using their published descriptions:[32],[93],[112],[134],[28],[119],[103],[115], [122],[106],[54].

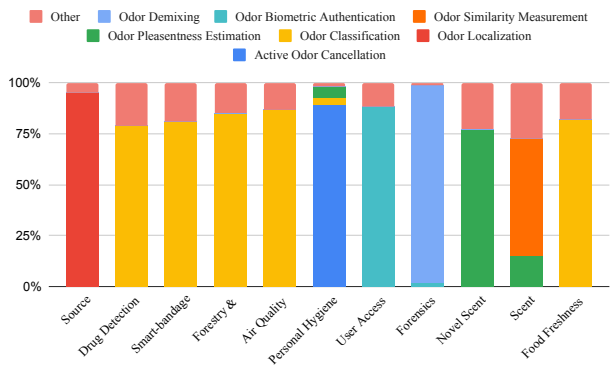


Figure 5: Breakdown of processing time spent by applications in each olfactory computing task. "Other" consists of time not spent in any specific olfactory computing task, and includes fetching and preprocessing inputs, initializing data structures, and storing results.

Table 2: Properties of computational kernels needed to support tasks for odor processing applications

Kernel	Input Size (Words)	Parameters	Avg. Linear Ops Between Non-linear Ops	Data Memory Required (Words)	% FLOPS Auto-vectorizable
AngleDist	6	0	12	6	89
Conv2D	32x32	3x3	N/A	1924	77
GNB	6	54	36	12	0
GD	6	43	110	36	9
KMeans	6	1536	159	1828	89
KNN	2	34	34	19	24
LDA	6	21	110	12	47
MLP	6	163	N/A	19	87
OMP	6	120	45	262	35
SVM	6	126	23	30	7
PCA	6x200	0	14	2496	32
PCA-Inf	6	36	N/A	42	86
PF	2	64 (Particles)	9	4090	10

3 ANALYZING ALGORITHMS FOR OLFACTORY COMPUTING

Table 2 presents the implementation parameters we considered in this work for the different algorithms or kernels used to support olfactory computing tasks. To understand the computational characteristics of these kernels in context of olfactory computing, we rely on profiling. Initial kernel profiling was performed on an Intel Xeon E5-2680 v4. Profiling kernels were written in C++ and compiled using g++ 11.3.0. The PCA, LDA, and KMeans algorithms were implemented using the ALGLIB library [24], SVM was implemented using LIBSVM [30], and OMP was implemented with KL1p[49]. E-nose data used in the algorithms comes from a dataset of wine spoilage thresholds [46], using six MOS sensors per sample. The dataset contains 200 samples (a number in line with previous work [27, 31, 41, 53, 54, 88, 132]) with one of three labels.

Profiling was performed using Linux perf. To determine the average linear operations between non-linear operations, a custom analysis tool written with Intel PIN was used. Data memory requirements, shown in Table 2, are from the bare-metal RISC-V implementations of the algorithms used to evaluate Ahromaa's RISC-V core mode (Sec. 4).

We make several observations about the computational characteristics of the kernels that may impact any architecture we build for these workloads. First, none of the kernels are integer kernels[2] — all kernels require fixed-point arithmetic. Thus hardware support for fixed-point operations is necessary for energy efficient computation of olfactory computing workloads.

Second, none of the kernels require complex numbers — this is very different than audio and image sensor processing which makes copious use of classical frequency domain signal processing algorithms [82, 90]. Unlike hearing and sight, which arise from sensing wave-based phenomena (i.e., light and pressure waves), the sensation of odor arises from sensing non-wave chemicals. Thus, there is no frequency or phase component to the sensed odor signal. Principal component analysis (PCA) does include computing eigenvalues, but since correlation matrices are symmetric, the eigenvalues are necessarily real valued. Thus hardware support for complex arithmetic and fast Fourier transform is not required.

Third, the primary arithmetic primitives are multiplication and addition, often in the form of multiply-accumulates needed for evaluation of linear functions, as found in ANN, SVM, LDA, PCA, AngleDist, and OMP. Although many kernels use non-linear operations, these operations are often either multilinear (i.e., multiplication of variables rather than variables by fixed constant), or are limited in number relative to multiplications and additions, and they are temporally isolated between long sequences of linear and multilinear operations.[3] Thus, dedicated hardware to accelerate linear operations is likely to provide significant benefits for olfactory computing application workloads.

Fourth, there is a large spread in working set size among kernels, with some kernels having very small working sets which can be supported by a very small data memory, or even with a large register file, while several kernels require significantly more data memory. In fact, the largest kernel, PF, can be made to require drastically more memory by increasing the number of particles used in its Monte Carlo simulation, which can lead to better localization accuracy. This wide range in memory requirements suggests that a scalable memory organization is likely to provide energy and power benefits for olfactory computing applications.

Fifth, many of the tasks support vectorization, even auto-vectorization, which indicates that parallel architectures are likely to perform well on olfactory computing application workloads.

4 AHROMAA: AN ULTRA-LOW-POWER ARCHITECTURE FOR OLFACTORY COMPUTING

Olfactory computing is applicable in many application domains. However, in many domains, the light computational requirements of its algorithmic primitives means the attention of architects is little needed. XR headsets, able to power high resolution 3D graphics, for example, will have little trouble running olfactory computing tasks. There are, however, application domains with extreme constraints which make olfactory computing interesting to architects. Form factors such as wearables, (earrings, pendants, brooches, etc), bandages (e.g., for detecting infections), adhesives [61] (e.g., for body odor monitoring), packaging [5] (e.g., smart packages that

[2] quantization for ANNs is explored in Sec. 6.

[3] Due to the computational simplicity of reLU activations, we ignore these in analysis of non-linear vs linear interspersedness.

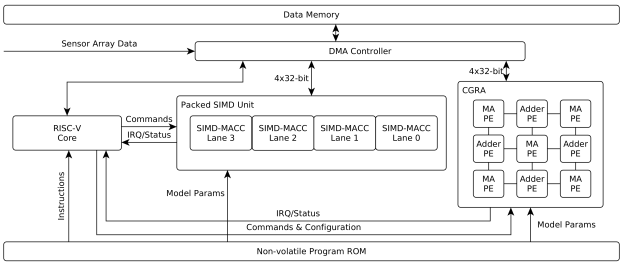


Figure 6: Organization of Ahromaa as a RISC-V core with packed SIMD unit and CGRA accelerators. Since Ahromaa aims for energy efficiency, achieved by operating at low voltage and frequency, Ahromaa exploits the lack of voltage scaling available to SRAMs to enable multiple SRAM accesses on a single port per clock cycle.

detect food spoilage), swabs (for breath, urine, body fluids-based diagnostics, for example), sensors-in-the-wild (e.g., air quality monitoring [1], surface and ground water quality monitoring, crop & forestry monitoring [2, 7]), etc., may only allow for energy harvesting. Field deployed sensors may need to maximize limited battery energy capacities over long periods, or be self-powered. In these domains, both power and energy are stringent limitations.

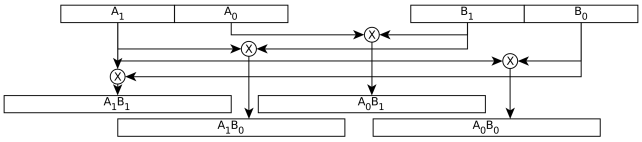
Therefore, as a case study, we consider the architecture of a programmable ultra-low-power olfactory computing hardware platform supporting low frequency and voltage operation; programmability is supported to lower costs and to maintain generality in a domain where applications are likely to evolve. Our architecture, Ahromaa (Fig. 6), is a heterogeneous reconfigurable odor monitoring and analysis architecture consisting of a 32 bit RISC-V MCU (RV32IM) with two integrated accelerators, both supporting fixed-point operations. The RISC-V core is a four-stage, in order, scalar core based off of the Open Hardware Group's CORE-V CV32 RISC-V IP [48]. Ahromaa can be used to support olfactory computing systems.

The first accelerator is a 128 bit packed SIMD unit which accelerates fixed-point fused multiply accumulate operations on 4, 8, 16, and 32 bit datatypes, to support the copious multiply-accumulates found in ANN, SVM, LDA, PCA, AngleDist, and OMP algorithms. Unlike a typical packed SIMD ISA extension, Ahromaa's packed SIMD unit does not have an additional SIMD register file. Instead, data is accessed directly from memory, which is made possible by the discrepancy between the core voltage at the minimal energy operating points of Ahromaa (Sec. 6), and the SRAM's nominal voltage required to ensure memory retention. Additionally, since the size of data memory required by many applications (Table 2) is small, the overhead of a dedicated SIMD register file may be large relative to data memory in its entirety (this is explored further in Sec. 6, where data memory is replaced/augmented with a flip-flop based register file). In order to exploit data reuse found in linear algebraic workloads (i.e., mapping a sequence of vectors by the same linear transformation), the packed SIMD unit contains a buffer which can be loaded with model parameters, or sensor data, neural network activations, and other runtime data.

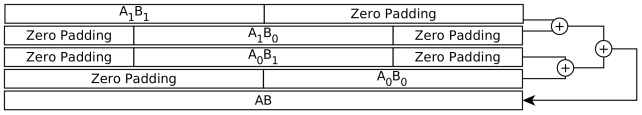
The second accelerator is a coarse-grained reconfigurable streaming dataflow architecture with nine heterogeneous PEs (adder PEs, and multiply-accumulate PEs) arranged in a 3×3 grid, and utilizing a bufferless, single-cycle, multi-hop routing network, as in [51, 123]. Each PE contains a small (16 B) buffer, in which it can store model parameters, as well as accumulated sums. Ahromaa's CGRA is based off of the SNAFU low-power reconfigurable architecture. PEs are capable of 32 bit width packed-SIMD execution of multiplies and adds (i.e., 1× 32 bit operation, 2× 16 bit operations, etc.). The dataflows supported on the CGRA enables mapping convolution layers with various kernel sizes and strides. This is important as although currently most olfactory computing ANNs are MLPs, there do exist CNNs for olfactory computing applications, and the choice of the best model for olfactory computing tasks is still being litigated.

Ahromaa's CGRA uses adders and multiply-accumulate PEs since olfactory kernels are dominated by multiplication and addition, often in the form of multiply-accumulates needed for evaluation of linear functions, as found in ANN, SVM, LDA, PCA, AngleDist, and OMP; GNB, PF, KMeans, etc., require adders. The number of adders and multiply-accumulates in the CGRA were chosen to limit Ahromaa power to 20 mW at nominal voltage and frequency. We used DSAGEN [124] to map dataflows onto the CGRA. To model a one-hop network in DSAGEN, our CGRA model in DSAGEN contains links between each PE; we added constraints to the dsa-scheduler to ensure that only mappings which do not violate the intra-PE link topology of Fig. 6 are considered. We chose 16 B buffers as these buffers support weight stationary dataflow for the Conv2D kernel, while also supporting input stationery and weight stationary dataflows for the linear transformations used in other kernels (SVM, LDA, MLP, etc).

Both accelerators support efficient packed SIMD addition and multiplication for datatypes of varying width, as wide adders and multipliers can be recursively built from narrow adders and multipliers. Wide adders are built from narrow adders via carry-out propagation. Wide multipliers are built from narrow multipliers via convolution (or, polynomial multiplication), as in Fig. 7. This figure shows how narrow multipliers can be used to compute partial products for a wider multiplier (Fig. 7a). These partial products are then summed to generate wider multiplier's product (Fig. 7b). When configured to support the 'narrow' datatype, the multiplier simply outputs the A_1B_1 and A_0B_0 products, while when configured to support the 'wide' datatype, the multiplier outputs the full AB product. This technique can be recursively applied, which enables 8 bit, 16 bit, and 32 bit multipliers to all be built from 4 bit multiplier primitives. Note that when operating on the 'narrow' datatype, there are twice as many partial product generating multipliers as required. These multipliers can be clock-gated, reducing dynamic power consumption in the multipliers and the adder reduction tree. A second use for these multipliers could be to increase the total width of the packed SIMD units. However, doing so would require doubling the data memory bandwidth. At ultra-low frequency operating points, this may be possible, however, it is not explored in Ahromaa. In fixed-point arithmetic, shifters are used to re-scale products. These are implemented for the different datatypes in parallel, and the output is multiplexed based on the datatype required. In both the packed SIMD unit and the CGRA PEs, the multipliers are pipelined to ensure they meet timing at nominal voltage and frequency. Exact location of pipeline registers is determined by register re-timing. To support ANNs, both accelerators are programmed to perform either linear or reLU activations on accumulated data before writing the results to data memory.



(a) Partial product computation.



(b) Partial product reduction.

Figure 7: Narrow multipliers are converted into wider multipliers via convolution, or polynomial multiplication.

The accelerators are given direct access to data memory, which enables the RISC-V core to be idle while the accelerators work. The accelerators signal completion of their work via IRQ lines. The core can also choose to poll the accelerator's status register if the accelerator is expected to finish its task quickly. Since accelerator launch costs are small (due to the small size of the CGRA, and small number of operations supported by both architectures), and the often large number of linear operations between non-linear operations (Table 2), the accelerators are focused on accelerating linear operations, and specifically fused multiply-accumulates. This means that non-linear operations are left for execution on the RISC-V core.

Ahromaa uses a modified Harvard architecture, in which program instructions and model parameters are stored in a non-volatile ROM, while read-write data is stored in an SRAM-based data memory. Since standard SRAMs cannot be safely voltage scaled without error detection and correction coding [73], a single SRAM bank may be accessed multiple times by the DMA controller in each compute clock cycle. Sec. 6 explores several different assumptions about the data memory organization, including scaling the sizes and number of banks, and replacing the SRAM with technologies which enable voltage scaling to be applied to the data memory.

Ahromaa may be run in one of three modes. In MCU mode, only the RISC-V core is active, with both accelerators powered off. In MCU+SIMD mode, the RISC-V core and packed SIMD unit are used, but the CGRA is powered off. In MCU+CGRA mode, the RISC-V core and CGRA are powered, while the packed SIMD unit is powered off. Which mode to use in deployment is determined statically by the programmer after analyzing performance in the different modes. In section 6, we obtain results for each olfactory algorithm in all three modes. Consequently, we consider Ahromaa's performance on a given metric to be that of the operating mode with the best performance on that metric.

5 METHODOLOGY

We evaluated Ahromaa's performance using cycle accurate RTL models. Power, area, and latency numbers are taken from analysis post synthesis in a commercial 65 nm technology. Such a legacy technode is appropriate for this work, as 65 nm process has low production and non-renewable engineering costs relative to more recent technodes. SRAMs were modeled with a commercially licensed 65 nm SRAM compiler, which provides SRAM area, as well as read and write power, and leakage power in normal, standby, and retention modes.

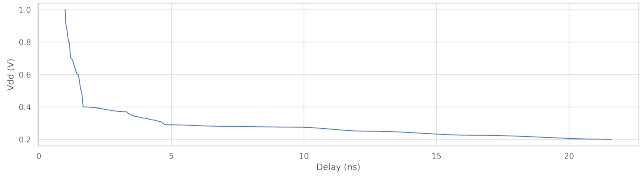


Figure 8: The supply voltage vs delay curves for an inverter chain made with high V_{th} transistors in 65 nm technology.

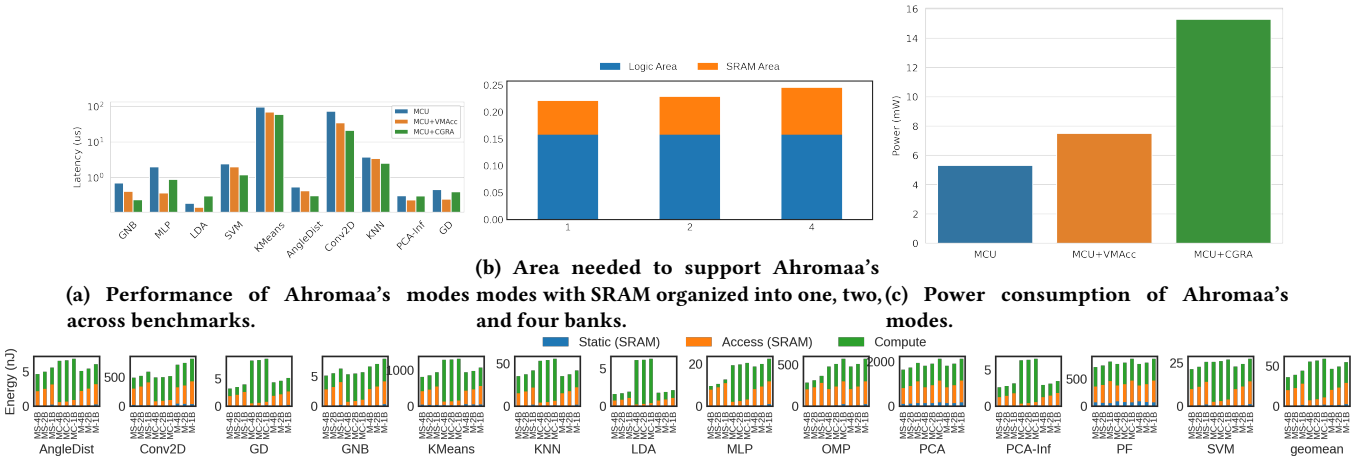
In order to evaluate the effects of voltage scaling on power and performance, we ran SPICE model simulations of voltage sweeps on inverters made from high V_{th} transistors, and recorded change in static and dynamic power consumption, as well as change in gate delay. The resulting Voltage vs Power and Voltage vs Delay curves (Fig. 8) were interpolated using cosine splines [92]. Previous work [55, 86] has shown that inverter chain-based power-delay scaling is valid for other logic. Indeed vast body of prior work has used inverter scaling for scaling power of complex circuits such as CPUs [55, 108, 133], GPUs [21], FPGAs [87], accelerators [89], etc. We use the same approach.

The kernels were implemented for the baseline MCU architecture. The kernels were written in Rust and compiled with rustc version 1.62. Linear algebra subkernels (e.g. matrix-multiply and symmetric eigen decomposition) were performed using the nalgebra [34] library.

6 RESULTS

Fig. 9 shows baseline results for Ahromaa with several SRAM organizations at nominal frequency (600 MHz) and voltage (1 V). Ahromaa's performance on the kernels is shown in Fig. 9a, broken down by Ahromaa mode of operation. Across all kernels, the mode with the packed SIMD unit consistently outperform the RISC-V core baseline (Fig. 9a). Similarly, for most kernels, the CGRA based mode outperforms the RISC-V core baseline. However, for applications with very small amounts of compute (e.g., LDA), the CGRA configuration and launch overhead offset the speed-up provided by the CGRA, resulting in faster execution on the RISC-V core alone. Despite this, the CGRA is the best performer on a number of benchmarks, including Conv2D - a kernel vital in CNNs, and SVM. The high performance of the parallel architectures is due to the prevalence of multiply-accumulates within the kernels, as discussed in Sec. 3.

The same pattern also emerges for energy (Fig. 9d), with the MCU+SIMD outperforming the MCU alone in energy efficiency, while the MCU+CGRA only outperforms the MCU on several kernels and outperform the MCU+SIMD on Conv2D, due to the high power consumption of the MCU+CGRA. The figure also shows the importance of using multiple SRAM banks. By dividing the SRAM into banks, banks which are unneeded for a given application kernel may be put into retention mode, eliminating nearly all power consumption of the SRAM.



(a) Performance of Ahromaa's modes across benchmarks.

(b) Area needed to support Ahromaa's modes modes with SRAM organized into one, two, and four banks.

(c) Power consumption of Ahromaa's modes.

(d) Total energy consumption of Ahromaa's modes for each kernel and with several SRAM organizations. 'M', 'MS', and 'MC' are for MCU, MCU+SIMD, and MCU+CGRA, respectively. The '[124]B' suffix gives the number of memory banks.

Figure 9: Baseline Results.

Fig. 10 shows the absolute energy consumed by Ahromaa's modes at their minimal energy operating point (MEOP) on a 10 mW nominal power budget. The MEOPs are 0.4 V and 357 MHz for the MCU and and MCU+SIMD, and 0.4 V and 119 MHz for the MCU+CGRA. Operating at the MEOP provides significant energy efficiency improvement over the nominal operating point due to reduction in dynamic power which decreases super-linearly with reduction in frequency. This can be seen by the growth in the proportion of energy consumed by SRAM reads and writes, relative to Fig. 9d. Operating at MEOP enables the MCU+CGRA to become the most energy efficient mode on most kernels, with MCU+SIMD being the most energy efficient on OMP and PF. On average, the MEOP provides 1.68, 2.79, and 1.49× improvement over the nominal operating points for the RISC-V core, RISC-V core+packed SIMD unit, and RISC-V core+CGRA modes, respectively.

Fig. 11 shows the effects of replacing SRAM with memories built out of flip-flops. In Fig. 11a, the entire 4 KiB SRAM is replaced by an equivalently sized memory built from flip-flops. This enables voltage scaling to be applied to the memory, whereas with a standard 6T-SRAM, scaling voltage down to the core's MEOP would likely result in data corruption. The effects of this are very positive for the MCU and MCU+SIMD modes, both seeing an over 60% decrease in energy consumption. In these modes, SRAM access energy dominates energy consumption. Thus, the lower energy of the flip-flop accesses leads to significant energy efficiency improvements. This comes despite the fact that the voltage-scaled memory inhibits the MCU+SIMD from operating at full throughput, due to voltage scaling of the flip-flop memory leading to increased access latencies. This fact also mitigates the effectiveness of this technique in the MCU+CGRA mode. Also, in this mode, compute energy, rather than memory access energy, is the predominant energy consumption.

In Fig. 11b, Ahromaa's memory is replaced by a small 256 B flip-flop based memory. For kernels with limited data memory requirements (which is the majority of the studied kernels), this memory organization results in significant (> 4×) energy savings in MCU and MCU+SIMD modes. Here, once again, the MCU+CGRA mode sees less improvement in energy consumption due to increased execution latency, and a smaller proportion of energy consumption being due to SRAM accesses in the baseline. Since the small memory consumes far less dynamic and static power than the 4 KiB memory, the improvement over the baseline grows, with the MCU+SIMD mode reducing power by over 80%. The smaller memory also helps MCU+CGRA mode, which sees a 30% reduction in energy consumption. Across all kernels, this data memory organization lead to additional 4.21, 4.37, and 2.85× improvements in energy efficiency on average.

The energy savings from flip-flop based memories come at a significant area cost. The area of Ahromaa with 4 KiB memory is 0.53 mm^2, a 193% increase over Ahromaa with an equally sized SRAM. The overhead of adding the 256 B memory to Ahromaa, however, is small. Using the small memory as a scratchpad, while retaining the standard SRAM for applications which may require more memory, results in an area increase of only 10.3%. This demonstrates that, in the the low-power olfactory computing domain, the small memory requirements of algorithms has a major impact on choice of computer architecture, with significant trade-offs between area (and hence cost) and energy efficiency. This trade-off can be mitigated by using hybrid memories, composed of small capacity, low density, low power memories, and larger, denser, higher power memories.

We also evaluate how parameter and activation quantization impacts energy efficiency in olfactory computing using the two ANN related kernels (MLP and Conv2D). Fig. 12 shows the effects of data quantization on the architectures at MEOP for a variety of different SRAM organizations. Benefits for the RISC-V core are the most limited, as the RISC-V core only takes advantage of quantization through reduction in number of memory accesses. The CGRA and packed SIMD unit also see benefit from reduced latency, as they are able to decrease the latency of the acceleratable portion of the computation via packed SIMD execution. Packed SIMD execution on multiplies and adds, as is needed by the packed SIMD unit and CGRA for the MLP and Conv2D workloads, has negligible hardware overhead, as word-length multipliers and adders can be built by recursively composing half-word, byte, and nibble length

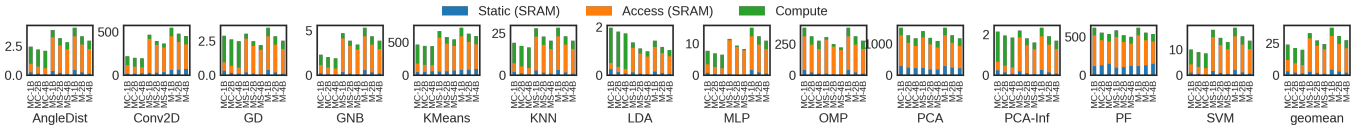
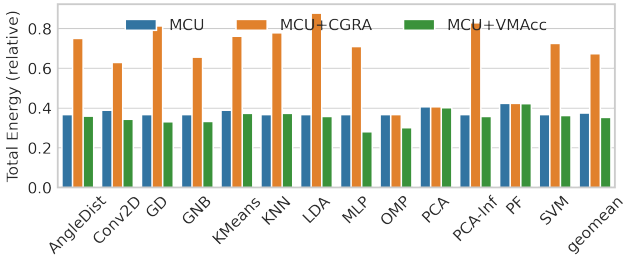
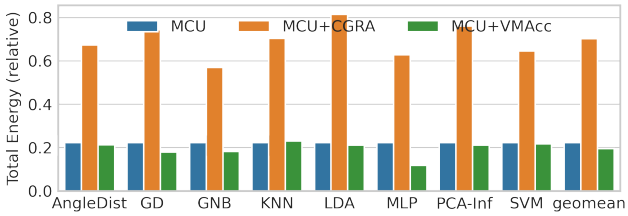


Figure 10: Energy consumption at the MEOP on each kernel with different SRAM organizations: One 2048 word SRAM, two 1024 word SRAMs, and four 512 word SRAMs.



(a) Energy achieved at MEOP for Ahromaa's modes across kernels with 4 KiB memories built out of flip-flops relative to execution with a standard 4 KiB SRAM.



(b) Energy achieved at MEOP for Ahromaa's modes across kernels with a 64-word memory built out of flip-flops relative to execution with a standard 4 KiB SRAM.

Figure 11: Replacing SRAMs with flip-flops has a dramatic effect on total energy consumption for some of Ahromaa's modes of operation.

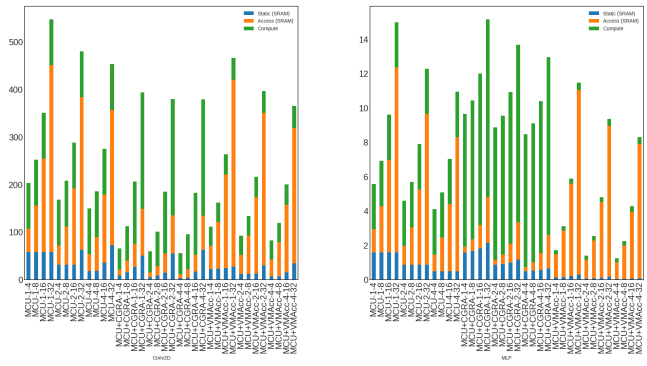


Figure 12: Data quantization leads to decreased energy for appropriate kernels. Further, data quantization's benefits are orthogonal to benefits from banking.

multipliers (via convolution) and adders (via carry propagation). Although not evaluated, the CGRA and packed SIMD unit could also see decreased dynamic energy consumption via input gating, as logic used to transform narrow (e.g., 4 bit) multiplications into wide (e.g., 32 bit) multiplications is not needed when performing quantized operations. Fig. 12 also shows that the improvements from quantization and from banking are sufficiently orthogonal of one another that they can be deployed simultaneously to achieve the effects of both without interference.

Indeed, there is also a *synergistic* relationship between banking and quantization, which can be most clearly seen in the Conv2D results for the RISC-V core mode. In this mode, since performance is independent of both banking and quantization, when the SRAM bank is available, the amount of SRAM leakage energy is fixed and independent of quantization. However, when two banks are used, since quantization leads to a decrease in the total amount of memory required to execute the Conv2D kernel, a single SRAM bank may be placed in retention for 4, 8, and 16 bit data, while both SRAM banks must be active for 32 bit data. The granularity of this again increases when four banks are available, with 4 and 8 bit data types only needing a single bank, while 16 bit data requires two banks, and 32 bit data requires all four banks. This same phenomenon plays out with the other modes. However, SRAM leakage energy in these cases is also impacted by quantization's impact on performance.

Ahromaa's results provide several interesting insights into olfactory computing. First, when area is not a concern (consider compute area on the back-side of a solar harvester), a programmable heterogeneous architecture will outperform monolithic architectures and should be used. Second, parallel SIMD and dataflow architectures are effective at improving energy efficiency of even the very small olfactory computing workloads — this is not self-evident, as many olfactory computing workloads have orders of magnitude less compute than even traditional sensor processing. Third, memory organization has a major impact on energy efficiency in olfactory computing, due to the wide ranging amounts of data memory required by its workloads.

Comparison against related work: There are works related to Ahromaa from the perspective of low-power design, including works on subthreshold sensor processors [86], other ultra low-power processors [76], and architectures for energy harvester powered processors [71]. While parallel architectures are often thought of as having high-power, recent works have proposed CGRAs which operate at far below nominal frequency [35, 51], enabling the energy efficiency associated with dataflow architectures, while still allowing ≈ mW operations. There has also been interest in low power SIMD architectures, including *scalar* vector processors [52], and vector processors for low-power media applications [50, 84].

Fig. 13 shows how a heterogeneous architecture such as Ahromaa outperforms an ULP MCU, an ULP MCU with a ULP SIMD unit, and a ULP CGRA (the individual operating modes of Ahromaa, respectively) by 1.36, 1.22, and 1.1× on average in energy efficiency

(in terms of energy at MEOP). This demonstrates the benefit of our architecture against possible alternatives.

Figure 13: Ahromaa's heterogeneous design allows it to outperform an MCU, an MCU with SIMD, and a CGRA baseline in energy efficiency at MEOP.

Fig. 14 shows the application level impact of Ahromaa - the entire application is mapped to Ahromaa for these results. On average, Ahromaa consumes 65%, 95%, and 77% of the energy used by the MCU, MCU+CGRA, and MCU+SIMD baseline architectures. Ahromaa benefits are due to its efficiency on olfactory computing kernels (Fig. 13), and because olfactory computing applications are dominated by these kernels (Fig. 5).

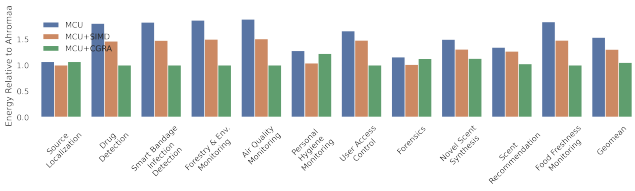


Figure 14: Ahromaa's energy efficiency vs an MCU, an MCU with SIMD, and a CGRA baseline in energy efficiency at MEOP on olfactory computing applications.

6.1 Ahromaa in a Full System

Consider a wearable patch used to evaluate body odor. One such system was prototyped in [135], with a six-sensor array of CNT based chemical sensors. However, the 'compute' in this prototype consists of an MSP430 MCU used *only* to transmit data from the sensor array to a nearby laptop where the olfactory computing actually occurs. Further, the prototype is powered by large batteries, impractical for a wearable patch. To commoditize such a wearable, olfactory computing should occur on the wearable, and the system should be self powered. Wearable piezoelectric harvesters are capable of harvesting power from the human body at densities up to 3.8 $\mu W\,mm^{-2}$, however, we assume a much more conservative 0.38 $\mu W\,mm^{-2}$ and a harvester with area 1 cm^2. Fig. 15 shows how long Ahromaa must sleep between runs of each algorithm in order to allow harvesting of sufficient energy. Results show that Ahromaa does not sleep for longer than 30 ms for any algorithm. This means that Ahromaa is often *not* the bottleneck on the system, as piezoelectric harvesters often run at < 5 Hz [45] to low hundreds Hz [58], and often have an energy production duty cycle well under $< 50\%$ [57, 97]. Thus Ahromaa can support the computational needs of energy constrained olfactory computing devices, such as wearable patches.

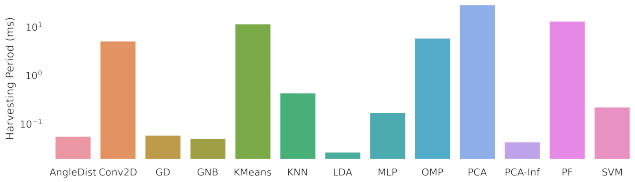


Figure 15: The energy harvesting period needed between runs of each algorithm on a olfactory computing system using Ahromaa and a 38 μW piezoelectric energy harvester.

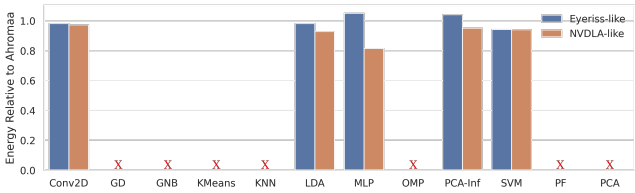


Figure 16: The kernel-level relative energy consumption of DNN accelerators vs Ahromaa. Kernels which cannot be mapped onto a DNN accelerator are marked with X

6.2 Ahromaa vs Deep Learning Accelerators

We also compared Ahromaa against an Eyeriss-like [33] and an NVDLA-like [136] DNN accelerator using the Accelergy-Timeloop architecture-level energy estimation methodology [129]. To ensure a fair evaluation between the accelerators and Ahromaa's CGRA, the accelerators were allocated equivalent memory (4 KiB) and number of PEs (nine in a 3×3 grid for Eyeriss-like, 3 PEs each with 3 LMACs for NVDLA-like). Timeloop-mapper was used to identify energy-optimal mappings using the compute and memory power and logic and memory energy data described in Sec. 5. Fig. 16 shows the energy consumption of the DNN accelerators relative to Ahromaa on Conv2D, LDA, MLP, PCA-Inf and SVM kernels (the remaining kernels feature significant amounts of computation not suitable for a DNN accelerator, and which cannot be mapped onto the accelerator substrates via Timeloop-mapper). Across the relevant kernels, the Eyeriss-like accelerator is within ±5% of Ahromaa. Surprisingly, the Eyeriss-like accelerator only beats Ahromaa by 2% on Conv2D. However, an architecture designed to exploit a weight-stationary data flow performs best on batched inputs, enabling fetched weights to be reused across multiple inputs, while the kernel evaluations assume a batch size of one, as is appropriate for an edge sensor processor. The NVDLA-like architecture outperforms Ahromaa by no more than 8% on all kernels but MLP. On MLP, the NVDLA-like architecture consumes only 82% of the energy of Ahromaa, using an input stationary dataflow.

Due to the limited performance improvement provided by the DNN accelerator, we chose not to include a DNN accelerator as part of Ahromaa's heterogeneous composition.

6.3 Generality of Ahromaa

Ahromaa is designed to support the characteristics of olfactory computing — small data, low data rates, and lax performance requirements. There are several other emerging domains with similar characteristics - e.g., sensing and measurement for light level [39],

Figure 17: Ahromaa leads to energy improvements on non-odor applications when those applications have data and performance characteristics similar to typical olfactory computing applications. Comparison is against the baseline MCU.

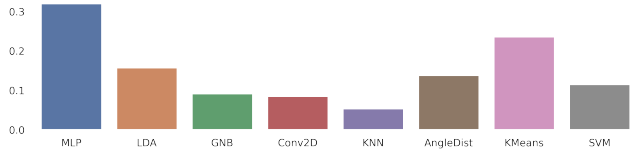


Figure 18: The kernel-level relative energy consumption of ASICs vs Ahromaa.

atmospheric temperature [39], noise level [13], barometric pressure [66], body temperature [98], heart rate [70], wind speed [25], nasal airflow [99], humidity [79], perspiration [91], blood pressure [75], step count [29], etc., which have low sample rate (subHz to a few Hz), sample precision (1-16 bits), and duty cycle (seconds to days). Ahromaa or similar olfactory computing systems could prove useful for these applications as well.

To demonstrate that Ahromaa can benefit these non-olfactory computing domains, we re-implemented strawberry health monitoring [39], and forest fire detection [79] applications with GNB classifiers, and an asthma emergency detection [98] application with an SVM classifier, and evaluated these applications using synthetic data on Ahromaa and the baseline four-stage RV32IM MCU. Results are shown in Fig. 17. Ahromaa provides $1.4 - 1.8\times$ energy improvements.

6.4 Future Research

Ahromaa is the first attempt at addressing the olfactory computing space. To quantify the opportunity for future work on olfactory computing hardware, we performed a simple limit study where we implemented kernel-specific accelerators in RTL and synthesized them in 65 nm technology. These ASICs differ from typical accelerators in that they target the exact kernel metaparameters, such as the number and shapes of layers in MLP, the filter size in Conv2D, etc, which enables optimizations beyond what are typically found in accelerators (e.g., the KNN accelerator includes an optimal switching network for the number of nodes in the dataset). Fig. 18 shows the energy consumption of the ASICs relative to Ahromaa. We see up to 19× benefit over Ahromaa motivating future research on logic and memory systems that can be configured to each kernel and reduced control overhead for fetching programmable instructions from memory.

Future research could also focus on alternative technologies that further trade off performance for power or support deeper voltage scaling. For example, a recent ultra-low power/performance EEPROM design [43] enables ROMs which consume 250 $\mathrm{pW\,bit^{-1}}$ leakage power at 90 nm (about 64x lower leakage than the 65 nm SRAMs considered in this work). This EEPROM, however, can only provide a read bandwidth of $< 2\,\mathrm{Mbit\,s^{-1}}$ — such EEPROMs could not support the memory bandwidth of Ahromaa, but could potentially support the memory bandwidth of a scalar architecture. Similarly, a recent22 nm 12T SRAM [69] supports $<$ nW $\mathrm{bit^{-1}}$ leakage and $<$ aJ $\mathrm{bit^{-1}}$ read and write energy at a reliable 0.2 V operating point. When scaled to 65 nm, the SRAM consumes $< \frac{1}{10}$ the leakage power/bit than a 65 nm flip-flop based memory, while using only $\approx 2\times$ the area of an 6T-SRAM memory.

Future work could also exploit the small size of several olfactory computing applications. These applications may be unrolled completely — i.e., turned into circuits — mitigating the need for memory altogether (modulo some IO buffering). Some of these applications also implement fixed models. For them, even the compute units can be optimized via constant propagation techniques, parameter-level quantization, etc.

Non-silicon electronics could also be interesting. Printed electronics enable bespoke customization at low cost [111]. Their poor performance (operating frequencies in the 10s-100s of Hz) may not be a liability in the olfactory computing domain due to its lax performance requirements. Similarly, flexible electronics [85] can also meet the performance requirements for olfactory applications while optimizing cost and conformality.

7 SUMMARY AND CONCLUSIONS

In this work, we argued that olfactory computing is an emerging domain for computer architects since a) a large number of driver applications have emerged, b) odor sensors have recently improved greatly, and c) non-traditional form factors such as wearable, electronic stickers, and xR devices that can host olfactory computing are already seeing wide adoption. Through a comprehensive review of olfactory computing literature, we identified a number of key olfactory computing algorithms which we profiled on existing hardware. Based on this profiling, we identified several characteristics, including the preponderance of fixed-point computation, and linear operations, and real arithmetic; a variety of data memory requirements; and opportunities for data-level parallelism. We proposed Ahromaa — a heterogeneous architecture for olfactory computing targetting extremely power and energy constrained olfactory computing workloads and evaluate it against baseline architectures of an MCU, a state-of-art CGRA, and a MCU with packed SIMD extension. Across our algorithms, Ahromaa's operating modes outperform the baseline architectures by 1.36, 1.22, and 1.1× in energy efficiency when operating at MEOP. We also showed how careful design of data memory organization can lead to significant energy savings in olfactory computing, due to the limited amount of data memory many olfactory computing kernels require. These improvements to the data memory organization lead to additional 4.21, 4.37, and 2.85× improvements in energy efficiency on average for all kernels.

8 ACKNOWLEDGMENT

We thank anonymous reviewers and Matthew Tomei for their feedback and NSF for partial support.

REFERENCES

[1] [n. d.]. Air Quality Monitoring System Market Drivers & Opportunities. https://www.marketsandmarkets.com/Market-Reports/air-quality-monitoring-equipment-market-183784537.html

[2] [n. d.]. Crop monitoring market size and trends forecast to 2025. https://www.marketsandmarkets.com/Market-Reports/crop-monitoring-market-8994590.html

[3] [n. d.]. Digital scent technology for modern scent marketing, aroma therapy and olfactory multimedia - aromajoin. https://aromajoin.com/products/aroma-shooter/

[4] [n. d.]. Figaro gas sensors - tgs1xxx and TGS2xxx Series / Figaro / gas / sensors -. https://en.maritex.com.pl/sensors/gas/figaro/figaro_gas_sensors_-_tgs1xxx_and_tgs2xxx_series.html

[5] [n. d.]. Food Packaging Market Size, share & growth report, 2030. https://www.grandviewresearch.com/industry-analysis/food-packaging-market

[6] [n. d.]. Head mounted display (HMD) market size, global forecast, growth drivers, opportunities 2030. https://www.marketsandmarkets.com/Market-Reports/head-mounted-display-hmd-market-729.html

[7] [n. d.]. Precision Forestry Market: Industry Analysis and Market Forecast to 2024. https://www.marketsandmarkets.com/Market-Reports/precision-forestry-market-228021988.html

[8] [n. d.]. Smartwatches market size, Share & Growth Report, 2030. https://www.grandviewresearch.com/industry-analysis/smartwatches-market

[9] [n. d.]. VR Smell technology to activate sensory stimulus. https://www.olorama.com/virtual-reality

[10] [n. d.]. Wearable Market Size, share & growth report, 2030. https://www.grandviewresearch.com/industry-analysis/wearable-technology-market

[11] 2004. The nobel prize in physiology or medicine 2004. https://www.nobelprize.org/prizes/medicine/2004/summary/

[12] 2019. Simulation and training. https://www.anthrotronix.com/our-work/simulation-and-training/

[13] Francesc Alías and Rosa Ma Alsina-Pagès. 2019. Review of wireless acoustic sensor networks for environmental noise monitoring in smart cities. *Journal of sensors* 2019 (2019).

[14] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. 2009. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning* 75, 2 (2009), 245–248.

[15] Alaa AlSlaity, Banuchitra Suruliraj, Oladapo Oyebode, Jonathon Fowles, darren steeves, and Rita Orji. 2022. Mobile Applications for Health and Wellness: A Systematic Review. *Proceedings of the ACM on Human-Computer Interaction* 6, EICS (2022), 1–29.

[16] Judith Amores and Pattie Maes. 2017. Essence: Olfactory interfaces for unconscious influence of mood and cognitive performance. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 28–34.

[17] Judith Amores, Robert Richer, Nan Zhao, Pattie Maes, and Bjoern M Eskofier. 2018. Promoting relaxation using virtual reality, olfactory interfaces and wearable EEG. In *2018 IEEE 15th international conference on wearable and implantable body sensor networks (BSN)*. IEEE, 98–101.

[18] Edoardo Aromataris and Alan Pearson. 2014. The systematic review: an overview. *AJN The American Journal of Nursing* 114, 3 (2014), 53–58.

[19] Artin Arshamian, Richard C Gerkin, Nicole Kruspe, Ewelina Wnuk, Simeon Floyd, Carolyn O'Meara, Gabriela Garrido Rodriguez, Johan N Lundström, Joel D Mainland, and Asifa Majid. 2022. The perception of odor pleasantness is shared across cultures. *Current Biology* 32, 9 (2022), 2061–2066.

[20] Alireza Bahremand, Mason Manetta, Jessica Lai, Byron Lahey, Christy Spackman, Brian H Smith, Richard C Gerkin, and Robert LiKamWa. 2022. The Smell Engine: A system for artificial odor synthesis in virtual environments. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 241–249.

[21] Prabal Basu, Hu Chen, Shamik Saha, Koushik Chakraborty, and Sanghamitra Roy. 2016. SwiftGPU: Fostering energy efficiency in a near-threshold GPU through a tactical performance boost. In *Proceedings of the 53rd Annual Design Automation Conference*. 1–6.

[22] VA Binson, M Subramoniam, and Luke Mathew. 2021. Discrimination of COPD and lung cancer from controls through breath analysis using a self-developed e-nose. *Journal of Breath Research* 15, 4 (2021), 046003.

[23] Andy Blanco-Rodríguez, Vicente Francisco Camara, Fernando Campo, Liliam Becherán, Alejandro Durán, Vitor Debatin Vieira, Henrique de Melo, and Alejandro Rafael Garcia-Ramirez. 2018. Development of an electronic nose to characterize odours emitted from different stages in a wastewater treatment plant. *Water research* 134 (2018), 92–100.

[24] Sergey Bochkanov. [n. d.]. https://www.alglib.net/

[25] Achilles D Boursianis, Maria S Papadopoulou, Panagiotis Diamantoulakis, Aglaia Liopa-Tsakalidi, Pantelis Barouchas, George Salahas, George Karagiannidis, Shaohua Wan, and Sotirios K Goudos. 2022. Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: a comprehensive review. *Internet of Things* 18 (2022), 100187.

[26] K Brudzewski, S Osowski, and W Pawlowski. 2012. Metal oxide sensor arrays for detection of explosives at sub-parts-per million concentration levels by the differential electronic nose. *Sensors and Actuators B: Chemical* 161, 1 (2012), 528–533.

[27] S Buratti, S Benedetti, M Scampicchio, and EC Pangerod. 2004. Characterization and classification of Italian Barbera wines by using an electronic nose and an amperometric electronic tongue. *Analytica Chimica Acta* 525, 1 (2004), 133–139.

[28] Alexandre Caron, Nathalie Redon, Patrice Coddeville, and Benjamin Hanoune. 2019. Identification of indoor air quality events using a K-means clustering analysis of gas sensors data. *Sensors and Actuators B: Chemical* 297 (2019), 126709.

[29] Vivek Chandel, Arijit Sinharay, Nasimuddin Ahmed, and Avik Ghose. 2016. Exploiting IMU sensors for IOT enabled health monitoring. In *Proceedings of the First Workshop on IoT-enabled healthcare and wellness technologies and systems*. 21–22.

[30] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 1–27.

[31] Quansheng Chen, Aiping Liu, Jiewen Zhao, and Qin Ouyang. 2013. Classification of tea category using a portable electronic nose based on an odor imaging sensor array. *Journal of pharmaceutical and biomedical analysis* 84 (2013), 77–83.

[32] Xin-xing Chen and Jian Huang. 2019. Odor source localization algorithms on mobile robots: A review and future outlook. *Robotics and Autonomous Systems* 112 (2019), 123–136.

[33] Yu-Hsin Chen, Tushar Krishna, Joel S. Emer, and Vivienne Sze. 2017. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks. *IEEE Journal of Solid-State Circuits* 52, 1 (2017), 127–138. https://doi.org/10.1109/JSSC.2016.2616357

[34] Sébastien Crozet. 2019. Nalgebra linear-algebra library: Nalgebra. https://www.nalgebra.org/

[35] Satyajit Das, Davide Rossi, Kevin JM Martin, Philippe Coussy, and Luca Benini. 2017. A 142mops/mw integrated programmable array accelerator for smart visual processing. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–4.

[36] S De Vito, E Massera, G Burrasca, A Di Girolamo, M Miglietta, G Di Francia, and D Della Sala. 2008. TinyNose: Developing a wireless e-nose platform for distributed air quality monitoring applications. In *SENSORS, 2008 IEEE*. IEEE, 701–704.

[37] Hossein Derakhshandeh, Sara Saheb Kashaf, Fariba Aghabaglou, Ian O Ghanavati, and Ali Tamayol. 2018. Smart bandages: the future of wound care. *Trends in biotechnology* 36, 12 (2018), 1259–1274.

[38] Tom Dewettinck, Kris Van Hege, and Willy Verstraete. 2001. The electronic nose as a rapid sensor for volatile compounds in treated domestic wastewater. *Water Research* 35, 10 (2001), 2475–2483.

[39] Savvas Dimitriadis and Christos Goumopoulos. 2008. Applying machine learning to extract new knowledge in precision agriculture applications. In *2008 Panhellenic Conference on Informatics*. IEEE, 100–104.

[40] Hamza Djelouat, Amine Ait Si Ali, Abbes Amira, and Faycal Bensaali. 2017. Compressive sensing based electronic nose platform. *Digital Signal Processing* 60 (2017), 350–359.

[41] Dongdong Du, Jun Wang, Bo Wang, Luyi Zhu, and Xuezhen Hong. 2019. Ripeness prediction of postharvest kiwifruit using a MOS e-nose combined with chemometrics. *Sensors* 19, 2 (2019), 419.

[42] Arnaldo D'Amico, Giorgio Pennazza, Marco Santonico, Eugenio Martinelli, Claudio Roscioni, Giovanni Galluccio, Roberto Paolesse, and Corrado Di Natale. 2010. An investigation on electronic nose diagnosis of lung cancer. *Lung cancer* 68, 2 (2010), 170–176.

[43] Igor V. Ermakov, Igor A. Zubov, Andrey V. Nuykin, Sergey A. Timoshin, and Evgeny S. Vasilyev. 2020. An Ultra Low-Power Low-Cost EEPROM for UHF RFID Tags in a Single-Poly 90 nm CMOS Process. In *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. 108–111. https://doi.org/10.1109/EIConRus49466.2020.9039064

[44] Hossein Rezaei Estakhroueiyeh and Esmat Rashedi. 2015. Detecting moldy bread using an e-nose and the KNN classifier. In *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 251–255.

[45] Tzeno Galchev, Ethem Erkan Aktakka, and Khalil Najafi. 2012. A piezoelectric parametric frequency increased generator for harvesting low-frequency vibrations. *Journal of Microelectromechanical Systems* 21, 6 (2012), 1311–1320.

[46] Juan C Rodriguez Gamboa, Adenilton J da Silva, Tiago AE Ferreira, et al. 2019. Electronic nose dataset for detection of wine spoilage thresholds. *Data in brief* 25 (2019), 104202.

[47] Julian W Gardner, Hyun Woo Shin, and Evor L Hines. 2000. An electronic nose system to diagnose illness. *Sensors and Actuators B: Chemical* 70, 1-3 (2000), 19–24.

[48] Michael Gautschi, Pasquale Davide Schiavone, Andreas Traber, Igor Loi, Antonio Pullini, Davide Rossi, Eric Flamand, Frank K. Gürkaynak, and Luca Benini. 2017. Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 10 (2017), 2700–2713. https://doi.org/10.1109/TVLSI.2017.2654506

[49] René Gebel. 2012. KL1p—a portable C++ library for compressed sensing.

[50] Joseph Gebis, Sam Williams, David Patterson, and Christos Kozyrakis. 2004. Viram1: A media-oriented vector processor with embedded dram. *DAC04* (2004), 7–11.

[51] Graham Gobieski, Ahmet Oguz Atli, Kenneth Mai, Brandon Lucia, and Nathan Beckmann. 2021. Snafu: an ultra-low-power, energy-minimal CGRA-generation framework and architecture. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 1027–1040.

[52] Graham Gobieski, Amolak Nagi, Nathan Serafin, Mehmet Meric Isgenc, Nathan Beckmann, and Brandon Lucia. 2019. MANIC: A Vector-Dataflow Architecture for Ultra-Low-Power Embedded Systems. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* (Columbus, OH, USA) *(MICRO '52)*. Association for Computing Machinery, New York, NY, USA, 670–684. https://doi.org/10.1145/3352460.3358277

[53] Antihus Hernández Gómez, Guixian Hu, Jun Wang, and Annia García Pereira. 2006. Evaluation of tomato maturity by electronic nose. *Computers and electronics in agriculture* 54, 1 (2006), 44–52.

[54] Silvia Grassi, Simona Benedetti, Luca Magnani, Alberto Pianezzola, and Susanna Buratti. 2022. Seafood freshness: e-nose data for classification purposes. *Food Control* 138 (2022), 108994.

[55] Scott Hanson, Bo Zhai, Mingoo Seok, Brian Cline, Kevin Zhou, Meghna Singhal, Michael Minuth, Javin Olson, Leyla Nazhandali, Todd Austin, et al. 2008. Exploring variability and performance in a sub-200-mV processor. *IEEE Journal of Solid-State Circuits* 43, 4 (2008), 881–891.

[56] Adam T Hayes, Alcherio Martinoli, and Rodney M Goodman. 2002. Distributed odor source localization. *IEEE Sensors Journal* 2, 3 (2002), 260–271.

[57] Jian He, Shuo Qian, Xushi Niu, Ning Zhang, Jichao Qian, Xiaojuan Hou, Jiliang Mu, Wenping Geng, and Xiujian Chou. 2019. Piezoelectric-enhanced triboelectric nanogenerator fabric for biomechanical energy harvesting. *Nano Energy* 64 (2019), 103933.

[58] Yuantai Hu, Huan Xue, and Hongping Hu. 2007. A piezoelectric power harvester with adjustable frequency through axial preloads. *Smart materials and structures* 16, 5 (2007), 1961.

[59] Nyayu Latifah Husni, Ade Silvia Handayani, Siti Nurmaini, and Irsyadi Yani. 2017. Odor classification using support vector machine. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE, 71–76.

[60] Muhammad Huzaifa, Rishi Desai, Samuel Grayson, Xutao Jiang, Ying Jing, Jae Lee, Fang Lu, Yihan Pang, Joseph Ravichandran, Finn Sinclair, et al. 2020. Exploring extended reality with illixr: A new playground for architecture research. *arXiv preprint arXiv:2004.04643* (2020).

[61] Fortune Business Insights. 2022. With 5.0% cagr, global adhesives and sealants market size worth USD 92.29 billion by 2029. https://www.globenewswire.com/en/news-release/2022/07/18/2480856/0/en/With-5-0-CAGR-Global-Adhesives-and-Sealants-Market-Size-Worth-USD-92-29-Billion-by-2029-Fortune-Business-Insights.html

[62] Lasse Jensen and Flemming Konradsen. 2018. A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies* 23, 4 (2018), 1515–1529.

[63] Sunil K Jha and Kenshi Hayashi. 2015. A quick responding quartz crystal microbalance sensor array based on molecular imprinted polyacrylic acids coating for selective identification of aldehydes in body odor. *Talanta* 134 (2015), 105–119.

[64] Sunil Kr Jha, Filip Josheski, Ninoslav Marina, and Kenshi Hayashi. 2016. GC–MS characterization of body odour for identification using artificial neural network classifiers fusion. *International Journal of Mass Spectrometry* 406 (2016), 35–47.

[65] Sunil Kr Jha, Ninoslav Marina, Chuanjun Liu, and Kenshi Hayashi. 2015. Human body odor discrimination by GC-MS spectra data mining. *Analytical Methods* 7, 22 (2015), 9549–9561.

[66] FJ John Joseph. 2019. IoT based weather monitoring system for effective analytics. *International Journal of Engineering and Advanced Technology* 8, 4 (2019), 311–315.

[67] Kathrin Kaeppler and Friedrich Mueller. 2013. Odor classification: a review of factors influencing perception-based odor arrangements. *Chemical senses* 38, 3 (2013), 189–209.

[68] Diclehan Karakaya, Oguzhan Ulucan, and Mehmet Turkan. 2020. Electronic nose and its applications: A survey. *International Journal of Automation and Computing* 17, 2 (2020), 179–209.

[69] Mehrzad Karamimanesh, Ebrahim Abiri, Kourosh Hassanli, Mohammad Reza Salehi, and Abdolreza Darabi. 2021. A robust and write bit-line free sub-threshold 12T-SRAM for ultra low power applications in 14 nm FinFET technology. *Microelectronics Journal* 118 (2021), 105185.

[70] Sufiya S Kazi, Gayatri Bajantri, Trupti Thite, et al. 2018. Remote heart rate monitoring system using IoT. *Techniques for Sensing Heartbeat Using IoT* 5, 04 (2018).

[71] Heewoo Kim, Javad Bagherzadeh, and Ronald G Dreslinski. 2022. SiC processors for extreme high-temperature venus surface exploration. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 406–411.

[72] Rajalakshmi Krishnamurthi, Adarsh Kumar, Dhanalekshmi Gopinathan, Anand Nayyar, and Basit Qureshi. 2020. An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques. *Sensors* 20, 21 (2020). https://doi.org/10.3390/s20216076

[73] Animesh Kumar, Jan Rabaey, and Kannan Ramchandran. 2009. SRAM supply voltage scaling: A reliability perspective. In *2009 10th international symposium on quality electronic design*. IEEE, 782–787.

[74] Grzegorz Łagód, Sylwia M Duda, Dariusz Majerek, Adriana Szutt, and Agnieszka Dołhańczuk-Śródka. 2019. Application of electronic nose for evaluation of wastewater treatment process effects at full-scale WWTP. *Processes* 7, 5 (2019), 251.

[75] Francesco Lamonaca, Eulalia Balestrieri, Ioan Tudosa, Francesco Picariello, Domenico Luca Carnì, Carmelo Scuro, Francesco Bonavolontà, Vitaliano Spagnuolo, Gioconda Grimaldi, and Antonio Colaprico. 2019. An overview on Internet of medical things in blood pressure monitoring. In *2019 IEEE international symposium on medical measurements and applications (MeMeA)*. IEEE, 1–6.

[76] Yoonmyung Lee, Yejoong Kim, Dongmin Yoon, David Blaauw, and Dennis Sylvester. 2012. Circuit and system design guidelines for ultra-low power sensor nodes. In *Proceedings of the 49th Annual Design Automation Conference*. 1037–1042.

[77] Hongyang Li, Bharat Panwar, Gilbert S Omenn, and Yuanfang Guan. 2018. Accurate prediction of personalized olfactory perception from large-scale chemoinformatic features. *Gigascience* 7, 2 (2018), gix127.

[78] Xin Li, Dehan Luo, Yu Cheng, Angus KY Wong, and Kevin Hung. 2020. A perception-driven framework for predicting missing odor perceptual ratings and an exploration of odor perceptual space. *IEEE Access* 8 (2020), 29595–29607.

[79] YongMin Liu, YiMing Liu, HongLei Xu, and Kok Lay Teo. 2018. Forest fire monitoring, detection and decision making systems by wireless sensor network. In *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 5482–5486.

[80] Patricio López, Roberto Triviño, David Calderón, Andrés Arcentales, and Ana V Guamán. 2017. Electronic nose prototype for explosive detection. In *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, 1–4.

[81] Panida Lorwongtragool, Enrico Sowade, Natthapol Watthanawisuth, Reinhard R Baumann, and Teerakiat Kerdcharoen. 2014. A novel wearable electronic nose for healthcare based on flexible printed chemical sensor array. *Sensors* 14, 10 (2014), 19700–19712.

[82] Danilo P Mandic, Soroush Javidi, George Souretis, and Vanessa SL Goh. 2007. Why a complex valued solution for a real domain problem. In *2007 IEEE workshop on machine learning for signal processing*. IEEE, 384–389.

[83] Patrick Mielle. 2022. A cold case: TGS 8xx gas sensors electronics. In *2022 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*. IEEE, 1–3.

[84] Byeong-Gyu Nam and Hoi-Jun Yoo. 2007. A low-power vector processor using logarithmic arithmetic for handheld 3d graphics systems. In *ESSCIRC 2007-33rd European Solid-State Circuits Conference*. IEEE, 232–235.

[85] Arokia Nathan, Arman Ahnood, Matthew T Cole, Sungsik Lee, Yuji Suzuki, Pritesh Hiralal, Francesco Bonaccorso, Tawfique Hasan, Luis Garcia-Gancedo, Andriy Dyadyusha, et al. 2012. Flexible electronics: the next ubiquitous platform. *Proc. IEEE* 100, Special Centennial Issue (2012), 1486–1517.

[86] Leyla Nazhandali, Bo Zhai, A Olson, Anna Reeves, Michael Minuth, Ryan Helfand, Sanjay Pant, Todd Austin, and David Blaauw. 2005. Energy optimization of subthreshold-voltage sensor network processors. In *32nd International Symposium on Computer Architecture (ISCA'05)*. IEEE, 197–207.

[87] Jose Luis Nunez-Yanez, Vassilios Chouliaras, and Jiri Gaisler. 2007. Dynamic voltage scaling in a FPGA-based system-on-chip. In *2007 International Conference on Field Programmable Logic and Applications*. IEEE, 459–462.

[88] Leiqing Pan, Wei Zhang, Na Zhu, Shubo Mao, and Kang Tu. 2014. Early detection and classification of pathogenic fungal disease in post-harvest strawberry fruit by electronic nose and gas chromatography–mass spectrometry. *Food Research International* 62 (2014), 162–168.

[89] Pramesh Pandey, Prabal Basu, Koushik Chakraborty, and Sanghamitra Roy. 2020. GreenTPU: Predictive design paradigm for improving timing error resilience of a near-threshold tensor processing unit. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 7 (2020), 1557–1566.

[90] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. 2011. Local laplacian filters: edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.* 30, 4 (2011), 68.

[91] Marc Parrilla, Tomàs Guinovart, Jordi Ferré, Pascal Blondeau, and Francisco J Andrade. 2019. A wearable paper-based sweat sensor for human perspiration monitoring. *Advanced Healthcare Materials* 8, 16 (2019), 1900342.

[92] HO Pascual, OA Fata, and AA Albanese. 2007. Impedance Measurement in Real Time, Employing Sine and Cosine Filters Simultaneously, Incorporating Spline Functions for Interpolation.. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)*. IEEE, 633–638.

[93] Nils Paulsson, Elisabeth Larsson, and Fredrik Winquist. 2000. Extraction and selection of parameters for evaluation of breath alcohol measurement with an electronic nose. *Sensors and Actuators A: Physical* 84, 3 (2000), 187–197.

[94] Dustin J Penn, Elisabeth Oberzaucher, Karl Grammer, Gottfried Fischer, Helena A Soini, Donald Wiesler, Milos V Novotny, Sarah J Dixon, Yun Xu, and Richard G

Brereton. 2007. Individual and gender fingerprints in human body odour. *Journal of the Royal society interface* 4, 13 (2007), 331–340.

[95] Adele Peters. 2019. The atmotube clips on your backpack to track the smog around you. https://www.fastcompany.com/90389662/clip-this-tiny-gadget-to-your-backpack-and-get-real-time-reports-of-the-air-quality-around-you

[96] Charles Platt. 1999. You've got smell! https://www.wired.com/1999/11/digiscent/

[97] Michele Pozzi and Meiling Zhu. 2012. Characterization of a rotary piezoelectric energy harvester based on plucking excitation for knee-joint wearable applications. *Smart Materials and Structures* 21, 5 (2012), 055004.

[98] A. Raji, P. Kanchana Devi, P. Golda Jeyaseeli, and N. Balaganesh. 2016. Respiratory monitoring system for asthma patients based on IoT. In *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*. 1–6. https://doi.org/10.1109/GET.2016.7916737

[99] Lydia Makarie Rofail, Keith KH Wong, Gunnar Unger, Guy B Marks, and Ronald R Grunstein. 2010. The utility of single-channel nasal airflow pressure transducer in the diagnosis of OSA at home. *Sleep* 33, 8 (2010), 1097–1105.

[100] Catherine Rouby, Benoist Schaal, Danièle Dubois, Rémi Gervais, and André Holley. 2002. *Olfaction, taste, and cognition*. Cambridge University Press.

[101] David A Shamma, Ryan Shaw, Peter L Shafton, and Yiming Liu. 2007. Watch what I watch: using community activity to understand content. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*. 275–284.

[102] Liang Shang, Chuanjun Liu, Yoichi Tomiura, and Kenshi Hayashi. 2017. Machine-learning-based olfactometer: prediction of odor perception from physicochemical features of odorant molecules. *Analytical chemistry* 89, 22 (2017), 11999–12005.

[103] Minglei Shu, Yunxiang Liu, and Hua Fang. 2014. Identification authentication scheme using human body odour. In *2014 IEEE International Conference on Control Science and Systems Engineering*. IEEE, 171–174.

[104] Aaron Smith. 2013. *Smartphone ownership-2013 update*. Vol. 12. Pew Research Center Washington, DC.

[105] Martin J Smith and Patrick J Kiger. 2006. The lingering reek of smell-O-vision. https://www.latimes.com/business/la-tm-oops6feb05-story.html

[106] Kobi Snitz, Adi Yablonka, Tali Weiss, Idan Frumin, Rehan M Khan, and Noam Sobel. 2013. Predicting odor perceptual similarity from odor structure. *PLoS computational biology* 9, 9 (2013), e1003184.

[107] Jose L Solis, Gary E Seeton, Yingfeng Li, and Laszlo B Kish. 2005. Fluctuation-enhanced multiple-gas sensing by commercial Taguchi sensors. *IEEE Sensors Journal* 5, 6 (2005), 1338–1345.

[108] Krishna Srinivasan, Ruhul Quddus, Russell K Mortensen, Herh Nan Chen, Manjunath Shamanna, and Po-loong Paul Chin. 2010. A core frequency-loss estimation methodology for a power-delivery network with reduced decoupling capacitance. In *19th Topical Meeting on Electrical Performance of Electronic Packaging and Systems*. IEEE, 53–56.

[109] Jude Stewart. 2022. VR still stinks because it doesn't smell. https://www.wired.com/story/vr-still-stinks-because-it-doesnt-smell/

[110] Martin Stokkenes, Raghavendra Ramachandra, and Christoph Busch. 2016. Biometric authentication protocols on smartphones: An overview. In *Proceedings of the 9th International Conference on Security of Information and Networks*. 136–140.

[111] Vivek Subramanian, Josephine B Chang, Alejandro de la Fuente Vornbrock, Daniel C Huang, Lakshmi Jagannathan, Frank Liao, Brian Mattis, Steven Molesa, David R Redinger, Daniel Soltman, et al. 2008. Printed electronics for low-cost electronic systems: Technology status and application development. In *ESSCIRC 2008-34th European Solid-State Circuits Conference*. IEEE, 17–24.

[112] Hao Sun, Fengchun Tian, Zhifang Liang, Tong Sun, Bin Yu, Simon X Yang, Qinghua He, Longlong Zhang, and Xiangmin Liu. 2017. Sensor array optimization of electronic nose for detection of bacteria in wound infection. *IEEE Transactions on Industrial Electronics* 64, 9 (2017), 7350–7358.

[113] Lina Sun. 2013. Liquid dangerous goods detection based on electronic nose odor recognition technology. In *International Symposium on Photoelectronic Detection and Imaging 2013: Infrared Imaging and Applications*, Vol. 8907. SPIE, 483–490.

[114] Bartosz Szulczyński, Tomasz Wasilewski, Wojciech Wojnowski, Tomasz Majchrzak, Tomasz Dymerski, Jacek Namiesnik, and Jacek Gebicki. 2017. Different ways to apply a measurement instrument of E-nose type to evaluate ambient air quality with respect to odour nuisance in a vicinity of municipal processing plants. *Sensors* 17, 11 (2017), 2671.

[115] Lidong Tan, Jiexi Wang, Guiyou Liang, Zongwei Yao, Xiaohui Weng, Fangrong Wang, and Zhiyong Chang. 2022. Model Development for Alcohol Concentration in Exhaled Air at Low Temperature Using Electronic Nose. *Chemosensors* 10, 9 (2022), 375.

[116] Mehmet Taştan and Hayrettin Gökozan. 2019. Real-time monitoring of indoor air quality with internet of things-based E-nose. *Applied Sciences* 9, 16 (2019), 3435.

[117] Jenny Tillotson, A Manz, and G Jenkins. 2006. Scent whisper. In *The Institution of Engineering and Technology Seminar on MEMS Sensors and Actuators 2006*. IET, 97–104.

[118] Binson VA, M Subramoniam, and Luke Mathew. 2021. Noninvasive detection of COPD and Lung Cancer through breath analysis using MOS Sensor array based e-nose. *Expert Review of Molecular Diagnostics* 21, 11 (2021), 1223–1233.

[119] Kush R Varshney and Lav R Varshney. 2014. Active odor cancellation. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*. IEEE, 25–28.

[120] J Včelák, P Ripka, J Kubik, A Platil, and P Kašpar. 2005. AMR navigation systems and methods of their calibration. *Sensors and Actuators A: Physical* 123 (2005), 122–128.

[121] Neil J Vickers. 2000. Mechanisms of animal navigation in odor plumes. *The Biological Bulletin* 198, 2 (2000), 203–212.

[122] Sorna Mugi Viswanathan and Revanth Rajan. 2020. Digital Scent Technology-A Critical Overview. *Int J Trend Sci Res Dev (IJTSRD)* 4 (2020), 218–221.

[123] Bo Wang, Manupa Karunarathne, Aditi Kulkarni, Tulika Mitra, and Li-Shiuan Peh. 2019. Hycube: A 0.9 v 26.4 mops/mw, 290 pj/op, power efficient accelerator for iot applications. In *2019 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 133–136.

[124] Jian Weng, Sihao Liu, Vidushi Dadu, Zhengrong Wang, Preyas Shah, and Tony Nowatzki. 2020. DSAGEN: Synthesizing Programmable Spatial Accelerators. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 268–281. https://doi.org/10.1109/ISCA45697.2020.00032

[125] Alphus D Wilson. 2013. Diverse applications of electronic-nose technologies in agriculture and forestry. *Sensors* 13, 2 (2013), 2295–2348.

[126] Alphus D Wilson. 2016. Recent progress in the design and clinical development of electronic-nose technologies. *Nanobiosensors in Disease Diagnosis* 5 (2016), 15.

[127] Chatchawal Wongchoosuk, Mario Lutz, and Teerakiat Kerdcharoen. 2009. Detection and classification of human body odor using an electronic nose. *Sensors* 9, 9 (2009), 7234–7249.

[128] Danli Wu, Dehan Luo, Kin-Yeung Wong, and Kevin Hung. 2019. POP-CNN: Predicting odor pleasantness with convolutional neural network. *IEEE Sensors Journal* 19, 23 (2019), 11337–11345.

[129] Yannan Nellie Wu, Joel S. Emer, and Vivienne Sze. 2019. Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8. https://doi.org/10.1109/ICCAD45719.2019.8942149

[130] Bin Yang and Wonjun Lee. 2018. Human body odor based authentication using machine learning. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1707–1714.

[131] Huichun Yu, Jun Wang, Cong Yao, Hongmei Zhang, and Yong Yu. 2008. Quality grade identification of green tea using E-nose by CA and ANN. *LWT-Food Science and Technology* 41, 7 (2008), 1268–1273.

[132] Huichun Yu, Jun Wang, Hongmei Zhang, Yong Yu, and Cong Yao. 2008. Identification of green tea grade using different feature of response signal from E-nose sensors. *Sensors and Actuators B: Chemical* 128, 2 (2008), 455–461.

[133] Bo Zhai, David Blaauw, Dennis Sylvester, and Krisztian Flautner. 2004. Theoretical and practical limits of dynamic voltage scaling. In *Proceedings of the 41st Annual Design Automation Conference*. 868–873.

[134] Xin Zhang, YuJie Qi, Xia Yang, HuiJuan Jia, et al. 2012. Evaluation of maturity of peach by electronic nose. *Journal of South China Agricultural University* 33, 1 (2012), 23–27.

[135] Yangong Zheng, Hanyu Li, Wenfeng Shen, and Jiawen Jian. 2019. Wearable electronic nose for human skin odor identification: A preliminary study. *Sensors and Actuators A: Physical* 285 (2019), 395–405.

[136] Gaofeng Zhou, Jianyang Zhou, and Haijun Lin. 2018. Research on NVIDIA Deep Learning Accelerator. In *2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*. 192–195. https://doi.org/10.1109/ICASID.2018.8693202