



A Unified Software/Hardware Scalable Architecture for Brain-Inspired Computing Based on Self-Organizing Neural Models

Artem R. Muliukov^{1*}, Laurent Rodriguez¹, Benoit Miramond¹, Lyes Khacef², Joachim Schmidt³, Quentin Berthet³ and Andres Upegui³

OPEN ACCESS

Edited by:

Anup Das,
Drexel University, United States

Reviewed by:

Debanjan Bhowmik,
Indian Institute of Technology Bombay,
India

Maryam Parsa,
George Mason University,
United States

*Correspondence:

Artem R. Muliukov
artem.muliukov@univ-cotedazur.fr

Specialty section:

This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: 30 November 2021

Accepted: 24 January 2022

Published: 02 March 2022

Citation:

Muliukov AR, Rodriguez L,
Miramond B, Khacef L, Schmidt J,
Berthet Q and Upegui A (2022) A
Unified Software/Hardware Scalable
Architecture for Brain-Inspired
Computing Based on Self-Organizing
Neural Models.
Front. Neurosci. 16:825879.
doi: 10.3389/fnins.2022.825879

¹ Université Côte d'Azur, Laboratoire d'Electronique, Antennes et Télécommunications, CNRS, Biot, France, ² Bio-Inspired Circuits and Systems Lab, Zernike Institute for Advanced Materials, Groningen Cognitive Systems and Materials Center, University of Groningen, Groningen, Netherlands, ³ Institute of Information Technologies, Hepia, University of Applied Sciences and Arts of Western Switzerland, Geneva, Switzerland

The field of artificial intelligence has significantly advanced over the past decades, inspired by discoveries from the fields of biology and neuroscience. The idea of this work is inspired by the process of self-organization of cortical areas in the human brain from both afferent and lateral/internal connections. In this work, we develop a brain-inspired neural model associating Self-Organizing Maps (SOM) and Hebbian learning in the Reentrant SOM (ReSOM) model. The framework is applied to multimodal classification problems. Compared to existing methods based on unsupervised learning with post-labeling, the model enhances the state-of-the-art results. This work also demonstrates the distributed and scalable nature of the model through both simulation results and hardware execution on a dedicated FPGA-based platform named SCALP (Self-configurable 3D Cellular Adaptive Platform). SCALP boards can be interconnected in a modular way to support the structure of the neural model. Such a unified software and hardware approach enables the processing to be scaled and allows information from several modalities to be merged dynamically. The deployment on hardware boards provides performance results of parallel execution on several devices, with the communication between each board through dedicated serial links. The proposed unified architecture, composed of the ReSOM model and the SCALP hardware platform, demonstrates a significant increase in accuracy thanks to multimodal association, and a good trade-off between latency and power consumption compared to a centralized GPU implementation.

Keywords: brain-inspired computing, neuromorphic architectures, self-organizing map, Hebbian learning, multi-modal classification, post-labeled unsupervised learning, FPGA, distributed computing

1. INTRODUCTION

1.1. Self-Organization as the Main Brain-Inspired Computational Principle

Nowadays, power consumption is one of the most crucial bottlenecks in the development of computing technologies. An important source of that inefficiency is a lack of specialization of computing systems for solving specific tasks, such as Artificial Intelligence (AI) problems. The best example of an effective system for solving equivalent problems is the human brain. Unlike their electronic counterparts, biological neurons use direct neural connections organized in a complex three dimensional structure. The signal of interneuron axons is transmitted in an extremely specialized manner, saving on the amount of transmitted information and permitting a high scalability of the system. This behavior appears completely different from the usual Von Neumann architecture found in a CPU. It offers universal computation capabilities but also has technical and intellectual constraints, mainly because of the physical separation between memory and computation units. This observation led us to develop a new computational paradigm combined with new electronic devices that support specialized neural models. Both are inspired by the self-organization property of the cortical areas of the biological brain, and their ability to learn their structure and function in an unsupervised manner while simultaneously acquiring data.

Self-organization can be defined as a global order emerging from local interactions (Heylighen and Gershenson, 2003) without a global controller or an external supervisor. In particular, local plasticity is the fundamental computational paradigm of cortical plasticity which enables self-organization in the brain, that in its turn enables the emergence of consistent representations of the world (Varela et al., 1991). In addition to biological plausibility, locality can be a key insight from the brain to enable online learning in embedded (and embodied) systems. In fact, local computing implies locality in time, which satisfies the real-time constraint of online learning, and locality in space, which is a direct consequence of the co-localization of memory and computation that satisfies the energy-efficiency constraint of on-chip learning. It is therefore a major shift from standard gradient-descend learning with back-propagation paradigm. Furthermore, it is unsupervised by nature and contrasts with supervised learning using labeled data, which is impractical in most online scenarios.

Therefore, the key inspiration of this work is the functioning of the cerebral cortex and its impressive self-organizing ability (Cain et al., 2016). Thanks to this ability, mammals and many higher animals create neural maps that represent their environment. Each map stores relevant information that, when combined, builds a model of the experienced objects and concepts. These maps can be built by the brain without the need for explicit data annotation or object labels. This property would be extremely useful for training on large amounts of data, where annotating itself becomes a difficult and expensive task. The Self-Organizing Map (SOM) (Kohonen, 1990) is one of the main unsupervised neural models based on such principles of cortical self-organization in the brain. It dynamically adapts to represent

the input data according to its distribution. This article explores the efficiency of the proposed brain-inspired computing software and hardware framework based on SOMs with application to the specific case of multimodal association problems.

1.2. Multimodal Association

Most processes and phenomena in the natural environment are expressed under different physical guises, which we refer to as different modalities. Multimodality is often considered as the first principle for the development of embodied intelligence (Smith and Gasser, 2005). Indeed, biological systems perceive their environment through diverse sensory channels: vision, audition, touch, smell, proprioception, etc. For example, we can recognize a dog by seeing its picture, hearing its bark or rubbing its fur. These features are different patterns of energy in our sensory organs (eyes, ears, and skin) that are represented in specialized regions of the brain. The fundamental concept is that there is a redundancy in neural structures (Edelman, 1987), known as degeneracy, which is defined as the ability of structurally different biological elements to perform the same function or yield the same output (Edelman and Gally, 2001). In other words, it means that any single function in the brain can be carried out by more than one configuration of neural signals, so that the system still functions after the loss of one component. It also means that sensory systems can educate each other, without an external teacher (Smith and Gasser, 2005).

The same principle can be applied for artificial systems, since information about the same phenomenon in the environment can be acquired from various types of sensors: cameras, microphones, accelerometers, etc. Due to the rich characteristics of natural phenomena, it is rare that a single modality provides a complete representation of the phenomenon of interest (Lahat et al., 2015). Therefore, an important problem in the development of modern AI systems consists in the aggregation of data from different sources to obtain a more complete and homogeneous understanding of the world, known as multimodal learning. In the brain, the representations built by different zones of the cortex might complement each other, for example by direct signal exchange (Cappe et al., 2009). This complementarity between sensory maps develops and relies on the self-organization mechanism of lateral synaptic communication. To simulate this mechanism, we use in this article Hebb's learning principle: "neurons that fire together, wire together" (Hebb, 1949). It suggests reinforcing the connections between co-activating neurons to capture the co-occurrence of specific data representations among different modalities caused by the same phenomenon.

Recent works have tried to study the human brain's ability to integrate inputs from multiple modalities (Calvert, 2001; Kriegstein and Giraud, 2006), and it is not clear how the different cortical areas connect and communicate with each other. To answer this question, Edelman proposed the reentry framework (Edelman, 1982, 1993). Reentry is a process which involves a population of excitatory neurons that simultaneously stimulates and is stimulated by another population (Edelman and Gally, 2013). For example, it has been shown that reentrant neuronal circuits self-organize early during the embryonic development of

vertebrate brains (Singer, 1990; Shatz, 1992), and can give rise to patterns of activity with Winner-Take-All (WTA) properties (Douglas and Martin, 2004). When combined with appropriate mechanisms for synaptic plasticity, the mutual exchange of signals amongst neural networks in distributed cortical areas results in the spatio-temporal integration of patterns of neural network activity. It allows the brain to categorize sensory inputs, remember and manipulate mental constructs, and generate motor commands (Edelman and Gally, 2013). Thus, reentry may be the key to multimodal integration in the brain. Based on the reentry paradigm, we have previously proposed the Reentrant SOM (ReSOM) model (Khacef et al., 2020b), a self-organizing artificial neural network based on local plasticity mechanisms for unsupervised learning and multimodal association (Khacef, 2020). In this work, we significantly upgrade it by defining its behavior for a scalable executing scenario and providing a first hardware implementation.

1.3. Contributions of the Paper

This work adopts the use of direct connections between SOMs to simulate the process of self-organization among areas of the cerebral cortex. This mechanism is proposed to solve basic AI problems such as clustering, classification, or anomaly detection. A distinctive feature of our algorithm is its scaling capability and the ability to train neural connections in the absence of direct data annotation. The architecture was tested in the presence of numerous data modalities, and the results of our model surpass the accuracy of previously published models (Khacef et al., 2020b; Rathi and Roy, 2021) by several percent, bringing us extremely close to real-life applications.

The model was implemented in the form of scalable constructors combining several digital devices (SCALP boards introduced in Section 4) connected through high-speed serial links (HSSL). Based on the results of our previous work (Khacef et al., 2020b), this article proposes an extension of the existing framework that enriches it with new tests and offers an implementation both in software for simulation and in hardware for real-time prototyping. This contribution presents a unified software/hardware architecture inspired by cortical self-organization and thus makes an important step toward embodied “brain-like” electronic systems. Such a model could be used in the future by an automated embedded system (robot, drone, autonomous car, space rover, etc.) to build a representation of its surrounding environment through multimodal sensors.

1.4. Outline of the Paper

The next section presents an overview of the state of the art related to the scientific questions addressed in this article. Section 3 describes the ReSOM model for multimodal unsupervised learning based on local computations. Section 4 gives general information about the SCALP board used for the hardware solution. Section 5 describes both software simulations on a CPU/GPU and the results of hardware deployment on FPGA-based boards. Section 6 briefly summarizes the tests conducted and discusses possible paths for future development.

2. STATE OF THE ART

2.1. Brain-Inspired Self-Organizing Neural Models

As observed in the brain, the product of the **self-organization** process is a representation of the stimulus that is feeding this structural self-organization, joining together the structure and the function. This ability to extract information from experience and interaction both with the environment and with others is observed especially in living systems. In our work we pay particular attention to analogous self-organizing models with the goal of reproducing the brain’s unique learning qualities. Regardless of the nature of the self-organizing model being considered, it must satisfy several properties that we seek to achieve for embodied brain-inspired computing:

- Capability of distributed computing (for hardware implementation and scalability purposes);
- Capability of unsupervised learning;
- Capability of multimodal data processing.

These specific behavioral needs point the way to another form of unsupervised learning compared to the approaches proposed in classical Machine Learning (ML). In this review, we do not go too deeply into the details of either the Auto-Encoders (AE) or the Semi-Supervised Learning (SSL) or Multi-Agent (MA) systems, as they involve relatively similar but at the same time quite distinct problems and restrictions compared to ours. We also do not try to cover all the existing self-organizing and unsupervised methods, considering the enormous number of publications in this domain. So, we mention here only some of the algorithms that have direct overlap with the unique features of our framework.

A distinctive feature of our model is the construction of an explicit data representation map of the objects. It is worth mentioning that other solutions tackling this issue have been proposed in the literature, although none of them meet all our criteria. One of them is the Elastic Map (Gorban and Zinovyev, 2005) method that creates a nonlinear distortion of the observed space, and another is the Generative Topographic Map (Bishop et al., 1998) that builds a map based on statistical principles, without an inspiration from neural systems. Separately, we note the Neural Gas (NG) model (Fritzke, 1995), which works on similar principles as the SOM, but allows more freedom in the neural structure. The NG model on the other hand, lacks the topological properties inherent in cortical regions, since its structure depends on the distance of each prototype from stimuli. This makes it complex to implement based only on local interactions of neurons, therefore complex to program in a distributed manner.

Also important in the context of self-organizing systems are Spiking Neural Networks (SNNs) based on local synaptic plasticity, such as Spike-Timing Dependant Plasticity (STDP) (Bichler et al., 2012; Diehl and Cook, 2015; Hazan et al., 2018; Rathi and Roy, 2021). These models exploit the temporal dynamics and the event-based computing of spiking neurons. Both SOMs and SNNs show interesting properties like **distributed computing** and **unsupervised learning**. But in

fact, even though the learning process of the SOM (Kohonen, 1982) and SNN (Diehl and Cook, 2015) can be distributed with local computing, the competition mechanism for the emergence of the “winning neuron” is usually implemented using either a centralized unit or an all-to-all connectivity between neurons. These two approaches are not scalable in execution time and connectivity, respectively, as further discussed in Khacef et al. (2019). In order to overcome this limit, we proposed in Rodriguez et al. (2018) to distribute the SOM computing based on the Iterative Grid (IG), a cellular neuromorphic architecture with local connectivity amongst neighbor neurons. This novel implementation gives a much better scalability of the SOM in terms of neurons. The IG is not used in this work, though, because our objective is to demonstrate the scalability of ReSOM in terms of SOMs, where every map represents a given modality.

This work is therefore based on the Kohonen’s SOM (KSOM), which is at a good level of biological abstraction for our application and has a number of advantages over the previously mentioned models. Typically, its advantages include all the previously cited desired properties, including ease of hardware implementation. Indeed, the presence of a predefined network structure endowed with a local neighborhood simplifies its further implementation in electronic circuits such as FPGA or ASIC. In parallel, it has been shown that SOMs perform better at representing overlapping structures compared to classical clustering techniques such as partitive clustering or K-means (Budayan et al., 2009). Recently, we showed that the SOM combined with transfer learning reaches a competitive accuracy on complex few-shot learning problems (Khacef et al., 2020a). In addition, SOMs were directly inspired by work on the cerebral cortex (Kohonen, 1990), and can hence be potentially applied in the future for the development of biologically compatible components.

The KSOM proposed more than thirty years ago (Kohonen, 1991) can serve as the main reference for our work, even though different adaptations of the original SOM model have since been proposed. We can distinguish several models offering some interesting and unique behaviors. The Dynamic Self-organizing Map (DSOM) (Rougier and Boniface, 2011), for example, dynamically adapts to changes in the dataset over time, continuously rebuilding the map structure if needed. The Growing SOM (GSOM) (Dittenbach et al., 2000) and the Plastic SOM (PSOM) (Lang and Warwick, 2002) can change the network structure by adding/deleting neurons during the learning process. So it becomes possible to dynamically adapt the size of the network to the data structure, but at the cost of a more expensive learning process. The Cellular SOM model (CSOM) (Girau and Upegui, 2019) also offers gains in accuracy and energy consumption, using a simplified grid to enhance its hardware implementation. The Pruning Cellular Self-Organizing Maps (PC SOM) model (Upegui et al., 2018) prunes some SOM connections to gain better performance. Furthermore, certain models such as the Semi-Supervised SOM propose to aggregate mixed supervised and unsupervised data to achieve better representations with the help of known labels (Braga et al., 2020).

Despite these possible extensions of the model, this work refers to the original KSOM version. This model is sufficient to prove the concepts addressed in this article. Moreover, we have shown that the standard KSOM method gives a better accuracy in classification tasks using post-labeled unsupervised learning (Khacef et al., 2019). Nevertheless, this choice does not exclude the possibility of using the unified framework proposed here with other versions of self-organizing cortex-inspired maps in future studies, depending on the problems being addressed.

2.2. Brain-Inspired Multimodal Frameworks

Multimodal interaction is another key aspect of this article. Taking into account the fact that each SOM is associated with the data of one modality, it can be assumed that each neural map acts as a region of the cerebral cortex. Our model uses Hebbian connections (Hebb, 1949) to transfer activity signals between two of these maps. By behaving in this way, the cortex is able to correct the weakness of one modality region using another. The proposed framework tries to achieve the same behavior, simulating the interaction among cortical zones of the human brain. The relevance of such a method is confirmed by studies in cognition that show the similarity of this training method to the developmental learning observed in children in the first years of life (Althaus and Mareschal, 2013). But to our knowledge, no computational model of this cognitive scheme has been proposed before. And moreover, no electronic version has been designed. The only work tackling this goal is ReSOM, a model proposed in Khacef et al. (2020b) on which this article is largely based. Specifically, this work is an extension of the ReSOM, keeping its general principles, but adding new features for scalability and a hardware implementation, as explained in the following.

A considerable number of previous works have been devoted to analyses of the interactions between self-organizing maps for the purpose of studying their communication (Lefort et al., 2010; Morse et al., 2010; Lallee and Dominey, 2013; Escobar-Juárez et al., 2016). Furthermore, most such studies have already tested the aggregation of several modalities, but with important differences from this work. In particular, the models were used to solve other problems, for example the ones found in developmental robotics (orientation of robots in space, sensory-motor coordination, etc.), and not the clustering problem targeted in this article. Also, the connection configurations were quite different and included an extra region for the modalities, resembling the so called Convergence-Divergence Zone (CDZ) (Lallee and Dominey, 2013). A distinctive feature of the CDZ-based method is the use of an additional map to connect all the SOMs together to combine their activities. The present work follows another paradigm and explores the retracing of direct reentry links between several neural maps, making it applicable to another type of AI problem (such as the clustering problem). Some authors propose a solution to the clustering problem (Jayaratne et al., 2021), but with a focus on the field of big data computing with map-reduction (map-reduce) approach. It uses the Apache Spark (Zaharia et al., 2016) framework and moves away from a biological inspiration.

Several other works in the literature have proposed to combine multimodal data without the direct use of self-organizing

mechanisms. These include the following: Cholet et al. (2019) proposed to use the Bidirectional Associative Memory; Parisi et al. (2017) proposed hierarchical architecture with Growing When Required (GWR) networks, where action-word mappings are developed by binding co-occurring audiovisual inputs using bidirectional inter-layer connectivity; Nakamura et al. (2011) proposed to use a joint probability with the Latent Dirichlet allocation model; and Vasco et al. (2020) proposed to enforce the classification by taking the Variational Encodings as a part of their framework. All these methods offer different points of view on the question of the representation of information in AI. They provide neither a midterm global framework which cognition tasks could rely on, nor an obvious short-term advantage over the model studied here. But we will monitor their evolution as possible competing models to our framework.

There is also a huge research area investigating multimodal encoders based on conventional neural networks. They achieve state-of-the-art results for solving particular machine learning problems. It is practically impossible to analyze all of them, but a significant number of them, at one step or another, use the classical concatenation of multimodal vectors (Chen et al., 2020; Xie et al., 2020; Bendre et al., 2021), without a deep examination of unique dependencies between them. Nevertheless, there are other models proposing smarter modality aggregation, such as the Contrastive Multimodal Fusion method (Liu et al., 2021), showing there is growing interest in the ML community for nontrivial multimodal fusion. However, they are still far from implementation in electronic circuits, especially for computation on separate and scalable electronic devices, as is proposed in this work.

Each method mentioned in this section may be placed in an overall compromise between performance and energy efficiency. It should be remembered that the energy cost of machine learning methods comes much more from the learning phase than from the inference phase (Brownlee et al., 2021). This aspect is often overlooked in even the most recent neural accelerators. Taking this issue into account from both a software and a hardware point of view is therefore essential for the more general deployment of AI in the future.

2.3. Neuromorphic Architectures

Neural networks implementations in the form of hardware architectures have gained an increased interest in recent years. The recent works around the Deep neural networks for pattern recognition have put a spotlight on neuromorphic engineering and some proposed architectures, like Paindavoine et al. (2015) or Pham et al. (2012) have demonstrated the computation efficiency of neuromorphic systems in terms of energy consumption. More generally, neuromorphic engineering aims to emulate as many neurons and synapses as possible with dedicated architectures in digital circuits as Schuman et al. (2015) or analog systems as Indiveri and Horiuchi (2011).

Leading projects in neuromorphic engineering have led to the creation of powerful brain-inspired chips capable of emulating multiple spiking neurons to investigate a new type of computing architecture to aid neuroscientists' research or pave the way for efficient embedded AI. For example, as part of

the DARPA's SyNAPSE project roadmap, the IBM's TrueNorth neuromorphic chip (Cassidy et al., 2013; Merolla et al., 2014) can implement 1 million digital neurons. The TrueNorth chip implements a very rich neural model, capable of reproducing many behaviors observed in biological neurons. As part of the European Human Brain Project, the SpiNNaker (Jin et al., 2010) project aims to model a billion biological real-time impulse neurons using a million ARM968 cores. More recently, Intel's Loihi neuromorphic chip (Davies et al., 2018) implements 130K Leaky Integrate and Fire (LIF) neurons and 130M synapses capable of online learning and inference. Its successor Loihi2 (Orchard et al., 2021) reaches 1 M neurons and introduces a more flexible microcode programmable neural engine that allows simulation of a wide range of different neural models. Though technologically impressive, these chips are designed for neural network simulation, not for the self-organization of hardware resources and are not intended to be used for SOMs models.

Self-organizing neural networks have also received their share of interest and several works have been published on the subject of dedicated SOM hardware architectures. As SOMs can be computationally expensive if implemented in a straightforward manner a lot of efforts have been invested to propose efficient dedicated hardware SOM architectures. Most of the existing works focus on optimizing the time and power performances by modifying the KSOM algorithm to new "hardware-friendly" variants, less expensive in terms of hardware resources such as logic and memory elements. Simplification of numerical representation (fixed point instead of float point) and/or redefinition of functions (Manhattan distance instead of Euclidean) are common optimization techniques aiming to optimize performance incurring in the cost of losing algorithm accuracy (Peña et al., 2006; Younis et al., 2009; Brassai, 2014; de Abreu de Sousa and Del-Moral-Hernandez, 2017a,b). More original vector codings have been proposed by Hikawa (2005) which proposes to encode the vector components as the duty cycle of square waveforms and to use Digital Phase-Locked Loops (DPLL) as parallel computing elements to measure and update the distances between vectors.

Other approaches focus on learning performance (expressed in millions of connection updates per second—MCUPS—per watt consumption). Lachmair et al. (2013) propose a bus-interconnected multi-FPGA hardware SOM that allows the implementation of large reconfigurable networks. Abady et al. propose a layered description of the HW SOM (Abadi et al., 2016, 2018) where the computational neurons are decoupled from the communication that is implemented with a NoC. The same idea is exploited in Jovanović et al. (2018) and Jovanović et al. (2020) in a more hierarchical manner as the interconnected computational cells no longer implement neurons but clusters of neurons. Each cluster computes a local winner and the global winner is computed through the NoC in a systolic way.

This interest is well-intended as efficient hardware architectures for self-organizing models will enable their application to autonomous and embedded systems, where unsupervised on-line learning is of first importance. Nonetheless,

there is no dedicated SOM architecture in the literature offering the flexibility required for direct multimodal SOM functioning.

3. REENTRANT SOM, A BRAIN-INSPIRED MODEL FOR MULTIMODAL UNSUPERVISED LEARNING

In this section, we describe the neural model, which enables the fusion of multiple modalities in a scalable and unsupervised manner. The model is an extension of the previously published ReSOM model (Khacef et al., 2020b) that only considered two modalities. Therefore, we start this section by defining the principles behind the KSOM, next we briefly discuss the training of intermodal (Hebbian) connections, and finally we put all these elements together to develop our neural framework for reentry multimodal interaction. This framework acts as an unsupervised method of clustering high-dimensional data that can then be used to solve several kinds of machine learning tasks. We finish this section with an application of the advanced ReSOM model to the specific case of classification tasks with numerous data sources.

3.1. Kohonen’s Self-Organizing Map

The model used here (Kohonen, 1991) is a neural network consisting of a rectangular 2D grid of neurons. Each neuron is endowed with a weight vector with the same dimension as the processed data. The neural network training process consists of the following stages:

- 0. Read a vector of multidimensional data.
- 1. Identify the closest neuron to the current input in the grid by computing and comparing L2 distance. In the following, this neuron will be called Best Matching Unit (BMU).
- 2. Train the network by changing the weights of the BMU and its neighbors in the direction of the data vector by the following formula

$$w_n(t + 1) = w_n(t) + \epsilon(t) \times h_\sigma(t, n, s) \times (v - w_n(t)) \quad (1)$$

where:

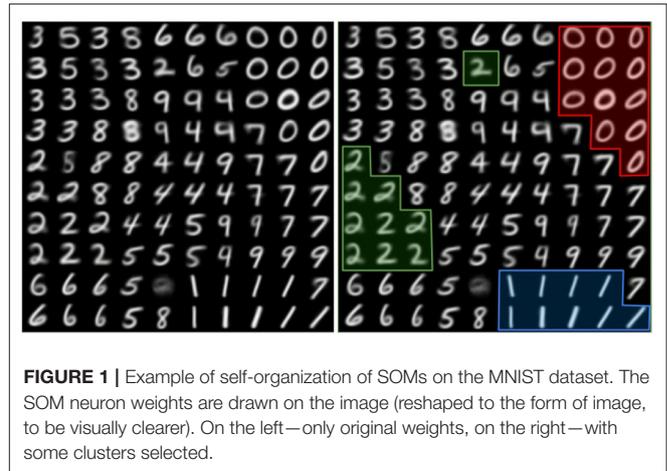
$$h_\sigma(t, n, s) = e^{-\frac{\|p_n - p_s\|^2}{2\sigma(t)^2}}; \quad (2)$$

$$\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{\frac{t}{T}}; \quad (3)$$

$$\epsilon(t) = \epsilon_i \left(\frac{\epsilon_f}{\epsilon_i}\right)^{\frac{t}{T}}; \quad (4)$$

With s —the BMU index in the grid, v —the data vector, and n —the index of any neuron. $\epsilon(t)$ and $\sigma(t)$ are decreasing functions to ensure the annealing of the learning; p_k —coordinate for a neuron with index k ; $h_\sigma(t, n, s)$ —the neighborhood function, t —“time” or learning iteration number. i and f refer to the initial and final values.

- 3. Repeat the operations (0, 1, and 2) for each data sample.



As a result, the grid of neurons forms a 2D representation of a real data space. Here, each neuron represents a group (or a class) of the previously shown objects, by being itself a typical (more precisely—averaged) data vector. Further division of the space of the representation neurons into known classes occurs, if necessary, when analyzing the associated labels. So, for example, the number 4 can be clearly represented as a closed and open figure (both in printing and writing styles). The drawing of clear boundaries between clusters is carried out in the next steps of the algorithm. But, for example in the simple MNIST case, as shown in **Figure 1**, the regions corresponding to the clusters attached to each number (“0,” “1,” etc.) can already be clearly observed.

3.2. Hebbian Connections

To model the interaction between the zones of the cortex, in this work it is proposed to use the model of pairwise connections between SOM maps. Each pair of neurons belonging to different SOMs is connected with a specific lateral synapse (as in **Figure 2**). During simulations, the weights of the connections between two maps are stored in a full-size matrix, with dimensions $d_1^2 \times d_2^2$, where d_1 and d_2 are the widths of each map.

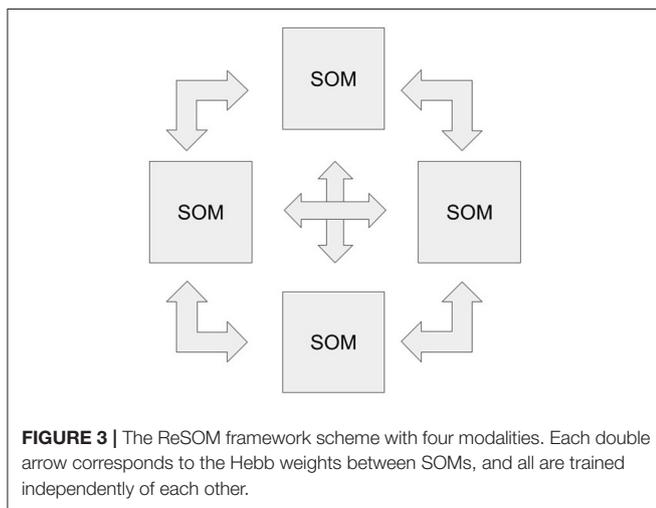
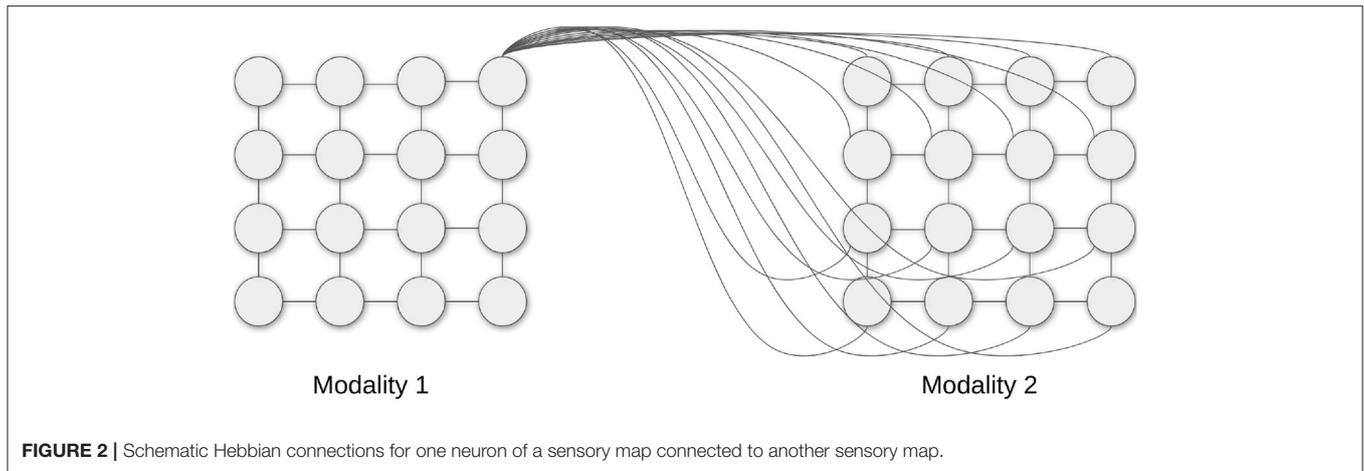
The Hebbian learning is conducted after that of individual SOMs. The Hebb principle is used to train the lateral weights. So the neural connection between two BMUs, with indices (x and y), belonging to different neural maps is strengthened according to the following law

$$W'_{BMU_x, BMU_y} = W'_{BMU_x, BMU_y} + \mu \times a^x \times a^y, \quad (5)$$

where μ is a hyperparameter and a^s is the maximal activation function coming from a SOM with index s (activation of th BMU):

$$a^s = \max_{n=0}^{k-1} e^{-\frac{\|v^s - v_n^s\|}{\alpha}} \quad (6)$$

Activations are counted and used for all existing vectors v_s , where s is the modality index. Thus, this operation must be repeated for all pairs of data samples representing different modalities.



3.3. ReSOM Structure and Learning

In order to scale the framework and be able to process more than two modalities, it is proposed in this work to:

- 1.1. Create the number of SOMs required to process all modalities (one map per modality).
- 1.2. Create the matrix of Hebbian weights for all pairwise compositions of all available SOMs (create $\frac{k(k-1)}{2}$ Hebbian connections for k maps). In **Figure 3**, an example of interaction between 4 modalities is depicted.
- 2. Train each SOM separately.
- 3. Using the received SOM activations, train all the $\frac{k(k-1)}{2}$ intermodal connections independently.

Thus, after training, we get a network of matrices representing the SOMs themselves and the connections between them, as in **Figure 3**.

3.4. Application of ReSOM to Classification Tasks

In order to make a quantitative analysis of our proposed model, we must use a quality metric of the resulting space clustering. To do that, we apply our framework to three subsets of data: a training set used for unsupervised learning composed of multimodal unlabeled data; a small annotated dataset used to attribute labels to neurons; and a testing set used for the validation and the evaluation of the clustering. This specific validation process leads at the end to the evaluation of a single metric corresponding to the accuracy of classification. That also means that this application consists in mapping the continuous space representation onto the discrete label space by assigning some class labels to the representation neurons. To do this, we use a certain amount of labeled data.

In fact, for any classification task based on unsupervised learning where the results need to be communicated in the so-called discrete label space, we need to find a method of attributing the identified clusters to existing labels. Different methods may be used to solve this problem, such as having an external expert who manually labels the obtained representations, or an automated process based on some labeled data. We use the latter approach, trying to minimize the number of required labels. The labeled data space should be large enough to represent all possible patterns of the available data. In practice, our results showed that only 1% of data for the MNIST dataset and 10% for the SMNIST dataset is enough to reach the best quality of classification (Khacef et al., 2020b). To have a uniform structure this work uses about 10% of labels for all the data modalities (precise numbers are given in Section 5). In the following sections, we discuss further the processes of labeling the neurons and testing and assessing the quality of a trained model.

Lastly, it should be noted that the problem addressed by our framework does not correspond to a typical SSL problem, since the existing labels are available (or provided) only after the training. So we consider a special type of SSL problem, called in the literature the Post-Labeled Unsupervised Learning problem (Khacef et al., 2020a,c).

3.4.1. Multimodal Labeling

The first step is the labeling process. In this work the objective is to deal with numerous modalities of data. The method is based on calculating the probability for each neuron to be assigned to a class. This probability is calculated as the sum of the neuron activations over all available annotated data for each separate label (as in Equation 7 below, for a SOM with index j and a neuron with index s). For each neuron, its activation consists of its afferent activity added to the activations coming from the lateral connections (Equation 8, where W are the weights of the lateral connections).

$$p(j, s, \text{label} = i) = \frac{\sum_{\text{label}=i} A_s}{\sum_{\text{label}=\text{any}} A_s} \quad (7)$$

$$A_s^j = c_1^j a_s^j + \sum_{\text{for } l \neq j} c_2^{lj} a^l \cdot W_s^{lj} \quad (8)$$

The coefficients c_1^{lj} and c_2^{lj} are model installation's hyperparameters, which may have values from 0 to 1. They set the relative importance of each modality during the labeling process, so these parameters may define a chosen labeling mode:

- If $c_2^{lj} = 0$ for **all** l and j , the mode is **unimodal labeling**. Here we label only using the individual SOM activation. This method was generally used in this work, as it is the simplest one;
- If $c_2^{lj} = 0$ for **some** l and j , the mode is **mixed labeling**. Here we label the neurons using their individual activations and some of the lateral activations;
- If $c_1^j = 0$ for **all** j , the mode is **divergence labeling**. Here we label the neurons of one SOM using the activations of the other modalities. For example we can label the audio modality using the data of the visual one. Previous studies have demonstrated that using this method we can reduce both the required quantity of annotated data and the number of annotated modalities without a loss in performance (Khacef et al., 2020b). In this article we do not explore this possibility, since optimizing the labeling is not the focus of this study;
- Other configurations may be used, but we do not discuss them all in this work.

3.4.2. ReSOM Testing and Accuracy Evaluation

Next, we discuss the method for testing the trained model. In the process of class prediction, we count the activations slightly differently than we do in the neuron labeling process. Here, the final ReSOM activation of each neuron consists of the product of its afferent activation and all the activations coming from the lateral SOMs, according to Equation (9).

$$A_s^j = a_s^j \times \prod_{l \neq j} a^l \cdot W_s^{lj} \quad (9)$$

$$j_{\max}, s_{\max} = \arg \max_{j,s} A_s^j \quad (10)$$

Next, we find the neurons, s_{\max} with the maximum activations among all the SOMs with index j using Equation (10). We propose to calculate the maximum activation among all available neurons of all available SOMs to choose the winning neuron (which gives the final prediction).

To evaluate the model we need to predict labels for all the test samples. The class of the selected neuron is compared with the ground truth. The accuracy is computed as the proportion of all the correctly predicted labels over the total number of test samples.

4. SCALP: SELF-CONFIGURABLE 3D CELLULAR ADAPTIVE PLATFORM

Network-on-chip is a natural evolution of the increasing complexity of system-on-chip architectures. In order to cope with prototyping requirements of such complex systems we have built a 3D multi-FPGA platform called SCALP (Vannel et al., 2018). SCALP is intended to provide flexible reconfigurability and 3D interconnectivity of its basic computation nodes. Each node is a PCB (printed circuit board) mainly containing an FPGA and HSSL-based connections that allow it to connect to its six neighbors (north, south, west, east, top, bottom). **Figure 4** shows an array of $3 \times 3 \times 3$ interconnected SCALP nodes.

Each SCALP node has a size of 10×10 cm and is implemented in a 12-layer PCB. Programs are executed by a Xilinx Zynq SoC (with a dual-core ARM Cortex-A9 processor @866 MHz and Artix-7 programmable logic with 74,000 cells), with its associated memory and enhanced communication capabilities permitting connections to neighbor modules through the HSSL with data rates up to 6.25 Gb/s.

A layered hardware/software architecture enables applications to be deployed on SCALP in a seamless manner. The ARM Cortex-A9 runs a Petalinux OS allowing users to write applications in C, C++, or Python. A set of libraries and drivers enables the use of services deployed on the FPGA. These services may be hardware accelerators or specific interfaces. In the case of the work presented in this article, they will permit access to the router built on the FPGA. **Figure 5** depicts the SCALP node internal architecture in the case of a 2D SCALP array (a third dimension is also supported). It is composed of two main layers, routing, and computation:

- The routing layer includes the equivalent of routing + link + physical layers in the OSI model, it guarantees link integrity between two neighbor nodes, and can redirect packets to neighbors in order to permit remote node communications without using intermediate computation nodes. A set of crossbar switches included in the FPGA supports the packet transmission.
- The computation layer is mainly composed of the processor running the application, its memory, and a Direct Memory Access (DMA) channel. The processor is in charge of handling application data in its local DDR memory and configuring DMA transfers for sending data to the routing layer. The interface between the router and the DMA is the same as the one between neighbor routers, built as an AXI



FIGURE 4 | Array of 27 interconnected SCALP nodes.

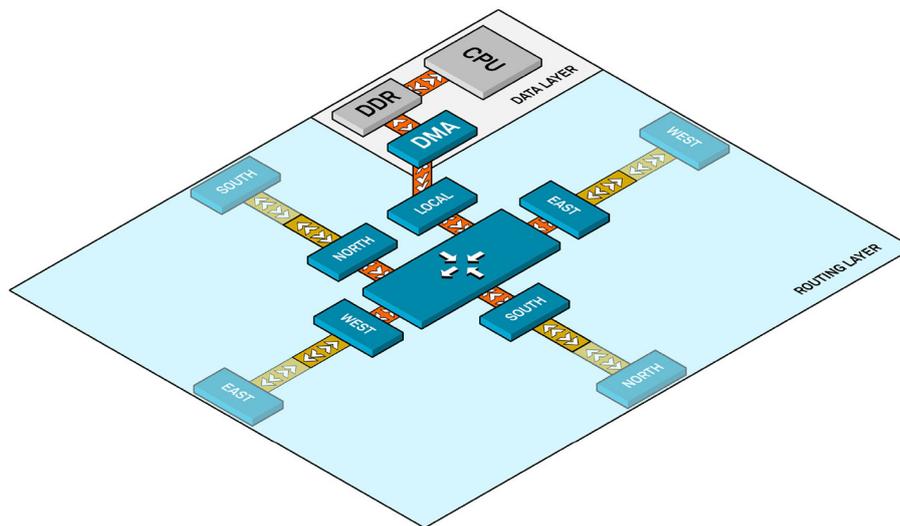


FIGURE 5 | SCALP node internal architecture for a 2D array.

stream interface. This uniform standard interface ensures the scalability of the system, in order to allow it to evolve to eventually add heterogeneous computation nodes.

SCALP is thus an excellent candidate for implementing ReSOM because its inherent cellular architecture permits a scalable deployment, given the distributed computing paradigm and the lack of hardware inter-dependency between nodes. Each of the SOM modalities can evolve in an independent SCALP board that adapts resources to its own computation needs. The communication is made transparent whatever the location of the connected maps, either local neighbor, or remote. The correlation between SOMs can thus be performed in a distributed and potentially asynchronous manner.

5. RESULTS

5.1. Experiments and Results

5.1.1. ReSOM Framework and Dataset Description

To test the model, a Python-based framework was implemented using the PyTorch library (Paszke et al., 2019) to speed up matrix calculations. Some simple datasets are used for the tests, such as the MNIST (LeCun and Cortes, 2010) and Spoken-MNIST (SMNIST) datasets. The SMNIST dataset is a subsample of Speech Commands (Warden, 2018) reduced to pronounced numbers from 0 to 9. The numbers were transformed to the Mel Frequency Cepstral Coefficients. The final version of the joint dataset in this configuration for two modalities was presented in a previous work (Khacef et al., 2020d).

TABLE 1 | Characteristics of the datasets used in the study.

Dataset	Vector size	Unique training objects	Unique test objects
Datasets table			
MNIST (LeCun and Cortes, 2010)	28 × 28	60,000	10,000
Spoken MNIST (Warden, 2018)	39 × 13	34,801	4,107
Fashion MNIST (Xiao et al., 2017)	28 × 28	60,000	10,000
Hand signs (Mavi, 2021)	28 × 28 (reshaped from 64 × 64)	1,531	515

The Fashion MNIST dataset (Xiao et al., 2017), representing images of different clothes, and the hand signs dataset (Mavi, 2021), representing numbers shown by hands, were used as the third and fourth modalities. Even if F-MNIST does not represent the same type of information as other considered datasets, it is one of the simplest datasets in the literature and it uses the same format as MNIST (10 classes and 28 × 28 images). It is assumed that each of the objects present in this dataset is arbitrarily associated to a specific class number between 0 and 9. More information about the dataset size can be found in **Table 1**.

All the datasets have been subsampled or oversampled (if needed) to have the same number of training/test vectors (60,000/10,000), but the data vector size is left unchanged. All the vectors have sizes comprised between 100 and 2,000 depending on the dataset used. We applied our labeling algorithm on a part of 5,000 training data vectors (about 8% of data were labeled). The ReSOM simulation code is available to the public at Muliukov (2021).

5.1.2. Preliminary Tests of ReSOM With Two Modalities

Due to the use of a new framework that accelerates matrix calculations, it was decided to return to the problem of two modalities, previously resolved with ReSOM. We check if the model improves with a wider search for the hyperparameters, keeping the old sizes of the SOMs (10 × 10 and 16 × 16 for MNIST and SMNIST, respectively). This time instead of a hard grid for such parameters as σ and ϵ , an advanced algorithm “Optuna” (Akiba et al., 2019) was used to optimize their values. Note that the search for hyperparameters can take a quite significant time (depending on using machines, code’s quality, etc.), which wasn’t evaluated in this work. Nevertheless, a few dozen runs are enough to find adequate (albeit not the best) hyperparameters.

As can be seen in **Table 2**, the accuracy results of the previous similar work (95.1%) have been achieved and even surpassed (with 96.6%). Notice that our model has used the datasets in a raw form, without any feature extractor, which might radically enhance the quality of the vectors representing the data. The addition of feature extractors as input of the neural maps increases the algorithm’s accuracy, as we showed in Khacef et al. (2020a,c).

TABLE 2 | Accuracy comparison of different methods processing two modalities.

Test name	Accuracy (%)
Unimodal	
MNIST (STDP SOM; Rathi and Roy, 2021)	93.2
T146 (stream audio) (STDP SOM; Rathi and Roy, 2021)	96.0
MNIST (Convolutional clustering unsupervised, 3,000 objects for labeling; Dundar et al., 2016)	98.6
MNIST (Deep Semi-supervised SOM, 10% of labels; Braga et al., 2020)	97.4
MNIST (Unsupervised STDP; Diehl and Cook, 2015)	95.0
MNIST (SOM—10 × 10, this work)	88.4
Spoken-MNIST or SMNIST (SOM—16 × 16, this work)	77.0
Two modalities	
MNIST+T146 (audio) (STDP Multimodal SOM; Rathi and Roy, 2021)	98.0
MNIST+SMNIST (ReSOM; Khacef et al., 2020b)	95.1
MNIST+SMNIST (ReSOM—this work)	96.1
MNIST+SMNIST (ReSOM + optuna—this work)	96.6

TABLE 3 | Comparison of accuracy up to 4x modalities with the ReSOM architecture.

Test name	Accuracy (%)
Conducted tests	
MNIST SOM (10 × 10)	88.4
Spoken-MNIST (16 × 16)	77.0
Fashion MNIST SOM (10 × 10)	78.47
Gests SOM (16 × 16)	55.63
MNIST+SMNIST	96.6
MNIST+SMNIST+FMNIST	99.0
MNIST+SMNIST+Gests	97.2
MNIST+SMNIST+FMNIST+Gests	98.36

Bold value indicates better accuracy.

5.1.3. Tests of ReSOM Scalability to More Modalities

The following tests presented in **Table 3** show the model’s scalability. We conducted tests of ReSOM with up to four modalities by adding the Fashion MNIST database (FMNIST) and a hand gesture database (denoted “Gests” in the following). It can be seen that the addition of new modalities can significantly increase the accuracy (up to the 99% in our tests). The best performance was achieved after optimizing the hyperparameters, details of which are available in a table in the article’s additional materials.

A deeper analysis may be done by analyzing the confusion matrices of each neural map (**Figure 6**). The use of the more modalities means each one helps to compensate for some of the mistakes of others. As an example let’s look at the prediction accuracy for the number “9.” The prediction accuracy of “MNIST + SMNIST” (**Figure 6E**) is relatively low (92%). But it reaches close to 99% (**Figure 6F**) using the extra capacities of FMNIST (**Figure 6D**), a modality which is quite good at predicting the number “9.” At the same time, adding the “Gests” modality (**Figure 6C**) was not as useful (**Figure 6G**—96%), possibly because of its lower accuracy in predicting “9” compared to the FMNIST modality.

Also we note that having greater number of modalities may even decrease the result (**Figure 6H**), either because of the

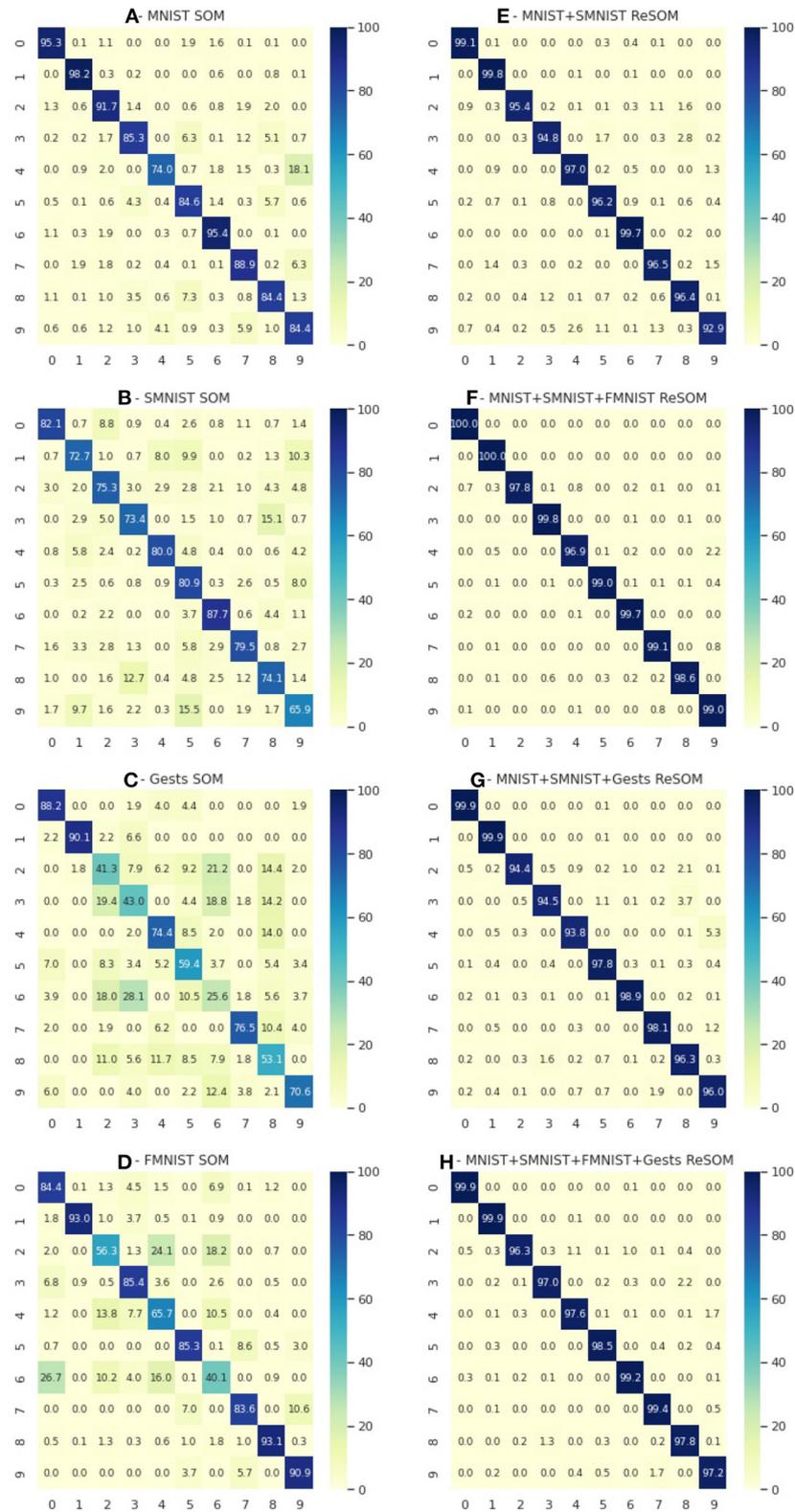
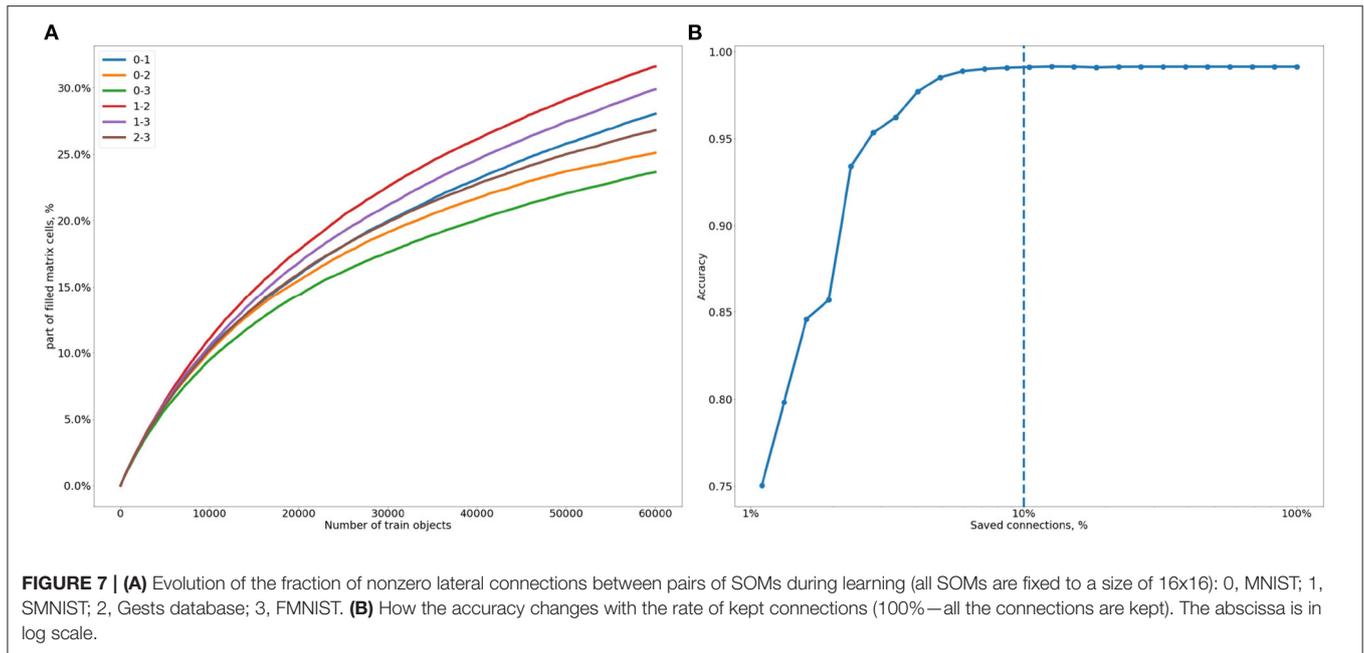


FIGURE 6 | The first column (A–D) is confusion matrices for SOM predictions for four datasets: MNIST, SMNIST, Gests and FMNIST. The second column (E–H) is ReSOM results for different combinations of datasets: “MNIST+SMNIST,” “MNIST+SMNIST+FMNIST,” “MNIST+SMNIST+Gests,” and “MNIST+SMNIST+FMNIST+Gests”.



addition of incompatible information (for lack of an adequate database) or due to the complexity of searching for the optimal hyperparameters of the model.

5.1.4. Connection Importance and Pruning

Reducing the number of values in the matrix can be extremely useful both for speeding up calculations and for reducing energy consumption in future implementations of the system. So next, we analyze the level of sparseness of the connections obtained and its influence on the resulting accuracy. The first thing to look at is the number of unused connections (i.e., those with zero weight). The fraction of such connections is quite large, as can be observed in **Figure 7A**, which shows the evolution of the number of nonzero connections during the training process.

We can observe some differences in the density of connections depending on the SOMs to be linked together. For example, the difference in the number of connections about of 1.3 times between the most linked pair of SMNIST and Gests database (“1–2”), and the less linked pair of MNIST and FMNIST (“0–3”). The difference in connectivity may, for example, be determined by the complexity of the represented modalities, which, in turn, may influence the frequency of activation of various neurons.

Also, we can observe that the training has still not reached a plateau—therefore, the limit of the created connections can be much higher than the 30% shown in **Figure 7A**. Thus, we can suppose that many more neurons may become connected over time. This level of sparsity makes storing the matrix in a compressed format not profitable and quite unjustified. But we still wish to avoid the unnecessary memory usage, so we explore how strongly we can increase the matrix sparsity without a decrease in accuracy.

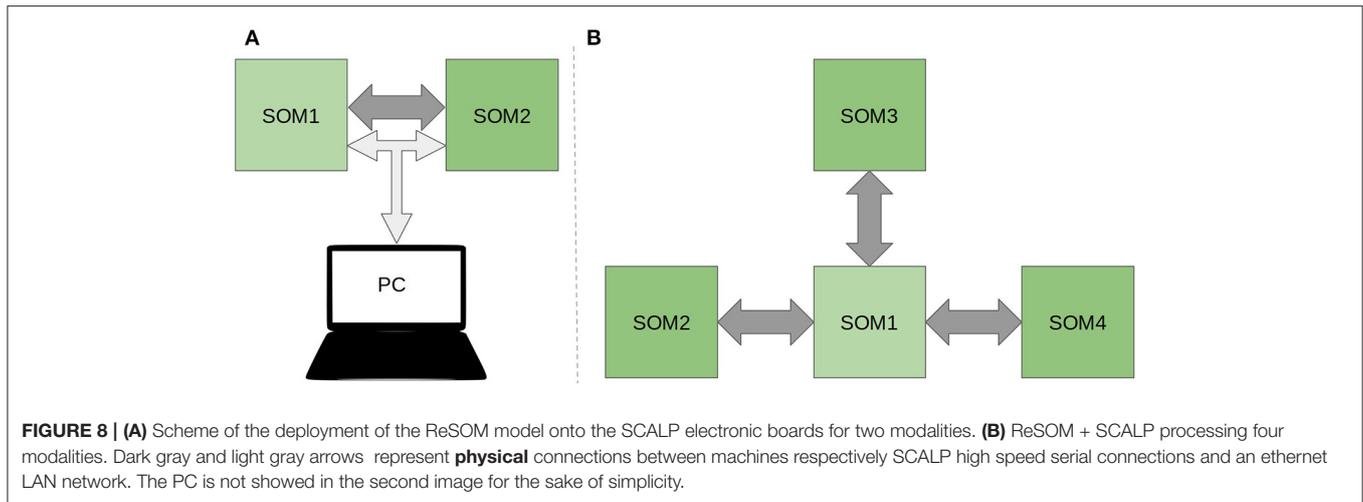
In fact, not all of the combinations have the same weight, so the influence of some might be much less than others. This leads us to the idea of cutting the weakest connections, that is, those attached to the smallest weight values. To search for the optimal threshold we track the influence of the number of smallest weights on the model’s accuracy, equating to zero a certain percentage of the weakest lateral connections. This information is plotted in **Figure 7B**. As long as more than 10% of the nonzero connections are retained, there is no significant effect on the accuracy. Thus, about 3% of matrix cells (10% of the average 30% cells filled among all matrices) are enough to achieve the maximal model accuracy.

5.2. Results of the Deployment of the Model on the Hardware Platform

5.2.1. General Description of the Experiment Scheme

To demonstrate the feasibility of the algorithm on real hardware, a simple scheme of interaction of two SOMs was implemented on the SCALP boards. The scheme involves two SCALPs passing data directly via the HSSL protocol. They are responsible for processing visual and auditory information and one of them also for the ReSOM prediction inference. The boards are connected to a PC in a local network via an Ethernet connection. The scheme can be seen in **Figure 8A**.

In the proposed scheme, the PC plays the role of program execution controller. It starts the different steps of the algorithm and controls the synchronization of data transfer between the devices. It also sends data vectors for testing and collects the final predictions, playing both the role of sensors (such as a camera) in a real system and the role of a monitoring system for the counted predictions.



Some parts of the algorithm (such as the calculation of SOM activations and the modification of the weights while training) are parallelized and so executed on different boards and computed independently of one another. These processes are controlled in different threads of the controller program written in Python and launched on the PC. The SOM calculations are executed on the boards as a Python server program. The nodes expose SOM and ReSOM computing functions as services that can be called through Python’s RPyC library. All requests are sent over the internal LAN network configured for standard communication between Bionic Beaver Linux on the PC and PetaLinux on the SCALP board.

For communication between the boards, direct connections with the HSSL protocol are used. Its schema allows the address of a node in the network to be set in three-dimensional coordinates and data to be sent to a desired address. In the current SCALP version, data are sent on a FIFO receiver of a board, where data packets of a size of 64 words of eight bytes are sequentially recorded. The packet sending function is a core application service and may be called via C/C++ code (or by a Ctypes call in Python). To send a data object of any type and size in Python, a program wrapper was written that serializes and splits files into the packets, and after that sequentially sends them to a given address. The receiver board reassembles them to the initial format.

5.2.2. Deployment Scenarios

Several deployment scenarios were tested to verify the global system performance. The first model to be tested was the inference model, to demonstrate the ability of a two-board system to produce values similar to those simulated earlier, using the already trained weights of both SOMs and lateral connections. The schema followed is shown in **Table 4**.

Further, in a similar way, the model was tested with the addition of the training of ReSOM lateral connections. The pipeline of information exchange is the same as the one presented earlier, except that it also contains an additional training step (see **Table 4**).

TABLE 4 | Board interaction pipeline for the ReSOM inference.

PC	Board 1	Board 2
Ask two boards to init themselves	–	–
–	Load MNIST weights, labels and ReSOM weights	Load SMNIST weights and labels
Confirm two boards init	–	–
Start of loop		
Send vector 1 to SOM1 and vector 2 to SOM2	–	–
–	Count activations of MNIST map	Count activations of SMNIST map
Confirm end of counting	–	–
Ask board 2 to send activations to board 1	–	–
–	Receive SMNIST activations	Send SMNIST activations
Confirm data received	–	–
Ask board 1 to make a prediction using ReSOM model	–	–
–	Count ReSOM predictions using two activation maps	–
Ask for and get the label prediction	–	–
End of loop		
Compute final accuracy and confusion matrix	–	–

5.2.3. Verifying the Functioning of the Hardware Framework

This test consists of verifying the model’s functioning on the hardware platform by comparison with the results of the previously tested PC simulation. The training of the entire ReSOM system is assumed to have been done offline on a PC. The weights are then fixed and the multimodal data are sent over the network to each neural map. So we compare two

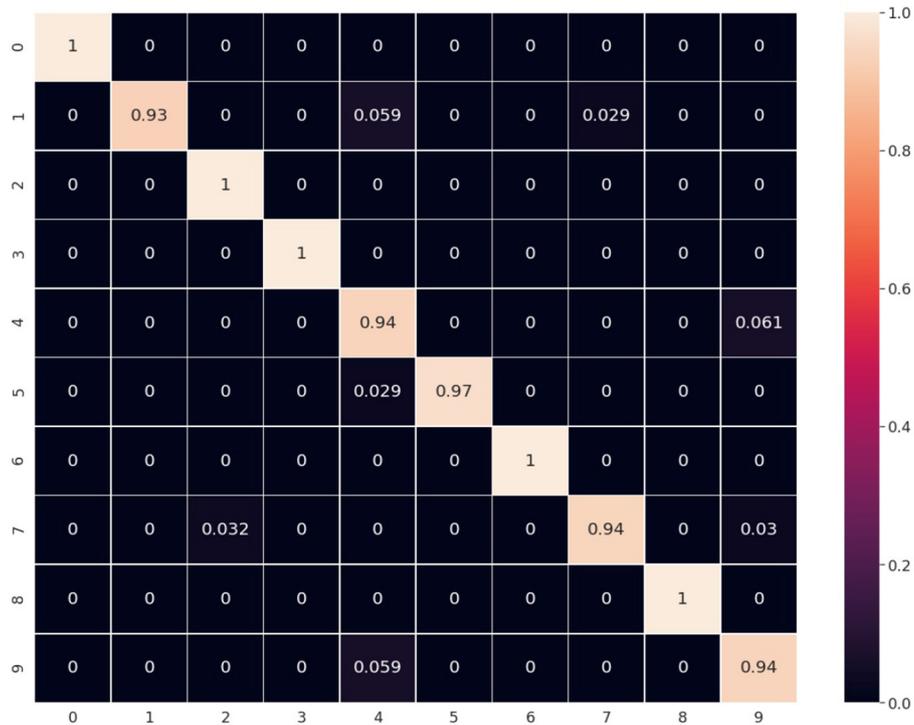


FIGURE 9 | Confusion matrix of ReSOM on tests conducted on SCALP boards. Accuracy = 0.966, test dataset size = 300 pairs of MNIST/SMNIST objects.

inference implementations to see if they work the same way on the same data.

Achieving the maximum model performance was not the focus of the research in this article; the main goal of the tests was to demonstrate whether such a model could be implemented. Because of some non-optimal features of the presented framework (which will be discussed later), we did not reach the maximum possible computational speed. Therefore, we decided to limit ourselves to a small test dataset (300 samples), sufficient to demonstrate the model’s functionality.

The accuracy of this on-board prediction test was equal to 96.6% and the resulting confusion matrix is presented in **Figure 9**. These results are exactly the same as the ones achieved by the simulation code, both in term of accuracy and the confusion matrix, thus confirming the validity of the deployed model.

5.2.4. Analysis of System Execution Performance

In the previous section, we conducted tests with a fairly small number of samples. This was due to the hardware architecture of the FPGA and non-optimized usage of the Python language. So the on-board execution for this first prototype does not yet reach the computation speed expected. But the use of multiple boards gives us the ability to perform parallel computations over the boards, which already gives a significant gain in system performance. The following section provides an analysis and numerical evaluation of the possible gain when scaling the system to more than two modalities. Another level of parallelism

TABLE 5 | Average execution times of inference steps.

Step #	Algorithm step	Time, s.
1	Count MNIST activations (SOM1, 10 × 10 size)	0.0795 ± 0.005
2	Count SMNIST activations (SOM2, 16 × 16 size)	0.1550 ± 0.005
3	Send activations from one board to another (matrix of 16 × 16 numbers)	0.016 ± 0.001
4	Make ReSOM prediction using two activations	0.011 ± 0.001
max(#1,#2) + #3 + #4	SCALP ReSOM on two boards	0.182 ± 0.01
#1 + #2 + #4	SCALP ReSOM on one board (calculated from the measured times)	0.246 ± 0.01

Bold value indicates better solution and lowest execution time.

consists in distributing the SOM map inside the FPGA circuit thanks to the Iterative Grid algorithm. This optimization will be considered in prospective work. As presented in **Table 5**, the calculation time of the SMNIST (neural map) activation matrix is an order of magnitude slower than its transfer time. Thus, computing in parallel and then transferring will result in a gain in system execution time over computing the algorithm on a single device. Adding more devices will enable further parallelization gains and therefore a greater increase in performance relative to executing on a single board. To demonstrate this, consider a model system composed of four SOMs implemented on four boards (**Figure 8B**).

TABLE 6 | Gain in execution time for scaled system.

Act. #	Alg. step	Time, s.
1	Count activations (approximate, same for all boards with same SOM of 16 × 16 cells)	0.16
2	Send activations from one board to another (hypothetical)	0.016
Data preparation for n modalities ReSOM		
(#1)+(n-1)*(#2)	SCALP ReSOM on n boards	0.16+(n-1)*0.016
n*(#1)	SCALP ReSOM on one board	n*0.1
Data preparation for four modalities ReSOM		
(#1)+3*(#2)	SCALP ReSOM on 4 boards	0.208
4*(#1)	SCALP ReSOM on one board	0.64

Bold value indicates lowest complexity and lowest execution time.

In this case, the parallel solution takes 0.208 s compared with 0.64 s for one board (as shown in **Table 6**); so the system gains a factor of 3 in the time needed to calculate and prepare the neuron activation matrices. This speedup is proportional to the number of boards in use. But we should note that the acceleration rate might decrease with the system's growth and may reach a state where it is impossible to directly connect all the boards to the one responsible for the ReSOM calculation.

5.3. Conclusion

5.3.1. Brain-Inspired Model

The cerebral cortex is capable of using self-organization for learning in the environment without the presence of embedded annotations. Individual cortical areas are responsible for processing different signals (in other words, modalities). Interaction between cortical zones allows the brain to build a complete picture of the world, reflecting a complex picture of interactions between different modalities using local computations.

This work combines simplified biological models to propose an architecture capable of similar behavior. ReSOM enables self-organization of the system and aids lateral connections between the “cortices” to exchange information, correcting weaknesses and information leaks among the different modalities. Our model is based on an interpretation of the interaction between real cortical zones. This fits perfectly with the idea of building a scalable system. Such a system was naturally implemented on microprocessor boards, allowing independent operation of several computational elements and improving the system's performance.

5.3.2. Implementation

The architecture allows the number of modalities to be scaled, and the computational efficiency will grow non-trivially by adjusting the number of SOMs. The model reaches an accuracy of 99% using only 3% of all possible ReSOM connections. This addresses a specific ML problem—the post-labeled unsupervised learning problem. Its advantage is the ability to determine the predicting system almost without any labels, partially solving the problem of expensive annotation.

The proposed model is implemented in hardware on SCALP boards. The possibility of scaling up the system to several modalities is also demonstrated. The use of high-speed serial connections allows information to be transferred directly between the boards, which gives an acceleration in system performance.

6. DISCUSSIONS

6.1. Limitations of the Current Prototype

6.1.1. Growth of the Number of Lateral Connections

The number of lateral connections, and the underlying Hebb computations, is a very important parameter that affects the memory consumption and execution time. To estimate the number of lateral connections for the ReSOM model, we can derive the following formula (11), which has a quadratic growth rate.

$$C_{ReSOM} = C_n^2 = \frac{n(n-1)}{2}; \quad (11)$$

For example, in the case of seven boards the ReSOM architecture will have 21 lateral connections, that is, three times more than the number of boards. So, for a small number of modalities, this growth is not so important, but as the number of modalities increases, it may affect the complexity of the signal transmission or the energy consumption.

An alternative method of multimodal aggregation is the convergence-divergence zone (CDZ)-based SOM combining all modalities in a single CDZ map. This method is not discussed in detail in this article, though we mentioned it in the literature review. To our knowledge, the ReSOM model currently offers higher accuracy than the CDZ. But, with a linear growth rate in the number of lateral connections, the CDZ-based SOM is a worthy candidate for developing a more scalable system, achieving a more impressive speed gain for a greater number of modalities.

6.1.2. Python Overhead

To speed up the prototype production and to simplify further integration with ready-made AI libraries, a significant part of the framework's code is written in Python. Unfortunately, non-optimized Python code entails a significant computation inefficiency and increase in execution time. This is largely due to the need to serialize, split into packets and then assemble the counted activations.

Such a decrease in the speed is not critical when the activation computation time significantly exceeds the transmission time. But in future implementations, with an increase in the calculation speed, this process will need to be optimized. This can be achieved either by rewriting the serialization-deserialization Python code in a low-level language as C/C++, or by reducing the limit on the size of forwarded packets. To solve this problem, we plan to implement a DMA interface between directly connected boards in a future SCALP system update.

6.2. Possible Development Directions

6.2.1. Hardware Development

A distinctive feature of SCALP is an integrated FPGA chip, which enables a grid of independent computing elements to be built. This article ignores this feature of the boards, but some solutions have previously been proposed (Girau and Upegui, 2019; Khacef, 2020). They develop the idea to use the FPGA for executing a SOM and accelerating the algorithm for calculating its BMU. Due to the direct exchange of information between neighboring SOM neurons, we can reduce the algorithm's complexity from $O(n)$ to $O(\sqrt{n})$, where n is the number of SOM neurons. In future, it is planned to correct this drawback and integrate one of the previously proposed Iterative Grid (Rodriguez et al., 2018) or Cellular SOM (Girau and Upegui, 2019) solutions in order to significantly speed up the calculations.

At this stage, in order to keep up with the speeding up of the SOM computations, the DMA implementation is much more important and useful, therefore this modification is already planned for the next SCALP release and will be integrated into our architecture.

6.2.2. Next Technical Algorithm Deployment

The obvious next step consists in testing the architecture's operation in the presence of 4-5 boards for processing different data modalities and comparing the performance of the ReSOM and CDZ methods for aggregating multimodal data in terms of accuracy, latency and power consumption. Growing the system up to more than a dozen boards makes sense to increase its performance, similar to the multi-thread solution proposed by Jayaratne (2021).

A further increase in the number of data modalities will be slowed down due to the difficulty of finding such a large number of information channels of a different nature. A possible way to solve this problem is to allow some of the boards to work with data of the same type, creating independent SOMs with data of the same nature. The development of such architectures will require both adaptation of the algorithm to work with several data aggregation nodes and new technical solutions for the implementation of the structure. This innovation could significantly boost the model's performance, so this extension is also planned for future implementations.

6.2.3. Online Learning, Acting, and Spiking Neural Networks

The ability to process signals and act in real time is an important feature of living systems that we have not discussed in this work. In future work, we will address aspects of our architecture that will allow it to develop into a full-fledged online acting agent. Some previously cited works (Lefort et al., 2010; Morse et al., 2010; Lalle and Dominey, 2013; Escobar-Juárez et al., 2016) show a significant potential for the development of self-organizing maps as a method for robots to navigate the surrounding space. The maps are able to model the spatial movement of agent and objects in space, so the multimodal connection can also help to capture the nature of spatial phenomena.

This model has no restrictions for processing information online and can also be used for real-time processing of incoming signals. This could be done by adding recurrence to the SOM (Voegtlin, 2002) or by integrating SNNs. SNNs have a distributed network structure (Xin and Embrechts, 2001; Ghosh-Dastidar and Adeli, 2009; Schliebs and Kasabov, 2013), but they do not forget the signal nature of received data and process them as spikes, in a sequential mode. Algorithms using SNNs have already allowed us to solve quite important and varied problems, such as unsupervised learning (Bohte et al., 2002; Dong et al., 2018), auto-encoding (Kamata et al., 2021), and even supervised AI problems (Kheradpisheh and Masquelier, 2020). Also, their good performance in terms of prediction accuracy and energy consumption (Amirshahi and Hashemi, 2019; Kim et al., 2020) promise their great potential for further development. The concept of SNN can be also combined with the SOM, as shown by other researchers (Hazan et al., 2018). Thus, we see great potential to develop our architecture by integrating it with SNNs for signal-type data processing.

The ReSOM multimodal association learning methods explored in this work were performed sequentially in two phases: first, we trained the SOMs for unimodal classifications, and second we created and reinforced bidirectional connections between pairs of maps based on their activities on the same training dataset. We refer to this learning approach as *asymmetric*. This is particularly interesting in the context of offline learning when working on pre-established datasets. First, in a purely practical way, it gave a lot of flexibility since we could train the unimodal SOMs on their respective available data separately, then train the multimodal association based on a smaller synchronized multimodal dataset. Synchronized here means that the multimodal samples that belong to the same class are presented at the same time. Second, from a developmental point of view, it has been shown that auditory learning begins before birth while visual learning only starts after birth (Althaus and Mareschal, 2013). Moreover, the ability to build associations between words and objects in infants appears to develop at about 14 months of age (Werker et al., 1998). The opportunity to process visual and auditory information sequentially may offer computational advantages in infant learning, as it could be a facilitating factor in the extraction of the complex structures needed for categorization (Althaus and Plunkett, 2015). These observations support the actual learning approach of ReSOM, where multimodal associations begin to develop after unimodal representations are learned sequentially.

Nevertheless, in the context of online learning in a dynamic and changing environment, another approach would be to perform both Kohonen-like and Hebbian-like learning at the same time, continuously. For example, this approach is followed using STDP learning in Rathi and Roy (2021). For this purpose, the KSOM would be replaced by a DSOM. The reason is that the KSOM has a decaying learning rate and neighborhood width, so that the learning stabilizes after a certain number of iterations. Therefore, the learning is stable but not dynamic, and can be considered as an offline unsupervised learning algorithm. In contrast, the DSOM is a variation of the KSOM algorithm where the time dependency of the learning rate and neighborhood

function has been replaced by a dependence on the distance between the BMU and the input stimulus. While the DSOM is less accurate than the KSOM (Khacef et al., 2019), it is more suitable for online learning. In addition, we would need a dynamic learning rate so that the multimodal association becomes stronger when the sample is well learned by the SOM, i.e., when the distance between the BMU and the sample is small. One way to do that is to use Gaussian kernel-based distances, so that the multimodal binding becomes more relevant after the convergence of the SOMs, without any manual tuning of the SOM hyperparameters.

6.2.4. Confronting SOMs to SNNs

Self-organization based on local plasticity mechanisms is a powerful computational principle that has been modeled in both SOMs and SNNs. In both cases, the basic computational unit is an artificial neuron whose synapses are plastic and learn to converge toward a “prototype,” i.e., the centroid of a given cluster of the data as shown in **Figure 1**. However, the SOM neuron and the spiking neuron computations are different, since they are at two different levels of abstraction from biological neurons. On the one hand, the SOM neuron is at a high level of abstraction where neural activity is modeled as a valued quantity, with a synchronous computation at each algorithmic time step which is adapted to frame-based sensors. On the other hand, the spiking neuron is at a lower level of abstraction where neural activity is modeled as spikes, and exhibits more biological plausibility (Maass, 1997). More importantly for practical applications, spike-based computation takes the timing of spikes into consideration, and consequently takes advantage of the spatio-temporal sparsity of event-based sensors streams with asynchronous computation. However, when applied to frame-based problems such as MNIST classification as in Diehl and Cook (2015), SNNs are inefficient because of the rate coding from images into spikes.

In fact, we have shown in a previous study that spiking neurons are less energy-efficient than formal valued neurons (a Perceptron was used as a reference) if three or more clock cycles are needed to make a prediction (Khacef et al., 2018). This is due to the integration and leak over time that have to be computed in several algorithmic time steps due to the spike coding from static frames. Therefore, SOMs are a better option for frame-based datasets, while SNNs have a greater potential for event-based datasets where time is inherently present in the patterns of interest and not an additional dimension due to the spike encoding. Both SOMs and SNNs have been applied to unsupervised learning as discussed in Section 2.1. Both networks rely on excitation and inhibition at the network level, with different inhibition mechanisms. In particular, the SOM uses a gradual inhibition in the topological neighborhood such that close neurons in the map learn the same patterns and converge toward similar prototypes. This property has been applied to SNNs in Hazan et al. (2018), resulting in a better classification accuracy.

SNNs have also been applied to multimodal association in Rathi and Roy (2021). In both SOM-based and SNN-based multimodal frameworks, every neuronal map receives a single afferent modality, representing a given cortical region.

Multimodal connections connect the different maps to improve the system’s perception of the environment which is quantified in these works as a classification performance. Nevertheless, the multimodal connections in the proposed multimodal SNN (Rathi and Roy, 2021) are uni-directional and initialized with a random connectivity. This limits the multimodal plasticity to connections that have been randomly fixed, which induces important variations in the network performance. In addition, it affects the system’s accuracy that goes down beyond 26% of the multimodal connections, while the ReSOM performance simply reaches a plateau of accuracy as shown in **Figure 7B**. A detailed comparison to the ReSOM model has been conducted in Khacef et al. (2020b).

6.2.5. Application to Other Problems

This work proposes one possible method for using lateral connections to transfer activations between SOMs. The model’s evaluation occurs using a non-negligible number of annotations (at least 1% of the training data). This is somewhat in conflict with the proposed self-organizing model, which is capable of completely unsupervised reasoning. However, the information stored in lateral connections may be rich enough to define the stable clustering on its own, with almost no use of the labels. Therefore, it seems possible to develop a completely unsupervised, or much less labeled algorithm, using graph cutting or distance clustering methods.

Such a system might be capable of learning on real data (such as video and audio signals captured simultaneously) by creating the clustering for all signal modalities using only the dependencies between them. Thus, simply by observing the objects around it, a robot could be capable of dividing the world into separate classes or categories. The model will thus be able to “understand” the world, by learning its distinguishable concepts. The development of such models is planned for future research studies.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

AM made the model’s simulation on a PC and implemented tests of the model on the SCALP electronic boards. He also wrote the majority of the text. LR implemented the library of information exchange between electronic boards in Python, actively participated in the discussions on the direction of the work. He wrote some parts of the text and contributed significantly to the editing. BM supervised the whole process of the work preparation, took an active part in the discussions about the direction of the work. He participated extensively during article’s writing. LK did preliminary tests of the model and took part in the planning of SCALP experiments. He also contributed to the article’s writing and editing. JS actively participated in the design of the SCALP

architecture, implemented the installation of the board's system and the programming of the messaging protocol between the boards. QB actively participated in the development of the SCALP architecture, configured the communication system between the boards, and participated on the test editing. AU supervised the implementation of the hardware part of the work. He took part in the discussions about the direction of the work, wrote some of the text and participated in editing the article. All authors contributed to the article and approved the submitted version.

FUNDING

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR)

REFERENCES

- Abadi, M., Jovanovic, S., Ben Khalifa, K., Weber, S., and Bedoui, M. H. (2016). "A scalable flexible SOM NOC-based hardware architecture," in *Advances in Self-Organizing Maps and Learning Vector Quantization*, eds E. Merényi, M. J. Mendenhall, and P. O'Driscoll (Cham: Springer International Publishing), 165–175. doi: 10.1007/978-3-319-28518-4_14
- Abadi, M., Jovanovic, S., Khalifa, K. B., Weber, S., and Bedoui, M. H. (2018). A scalable and adaptable hardware noc-based self organizing map. *Microprocess. Microsystems*. 57, 1–14. doi: 10.1016/j.micpro.2017.12.007
- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). "Optuna: a next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19* (New York, NY: Association for Computing Machinery), 2623–2631. doi: 10.1145/3292500.3330701
- Althaus, N., and Mareschal, D. (2013). Modeling cross-modal interactions in early word learning. *IEEE Trans. Auton. Mental Dev.* 5, 288–297. doi: 10.1109/TAMD.2013.2264858
- Althaus, N., and Plunkett, K. (2015). Timing matters: the impact of label synchrony on infant categorisation. *Cognition* 139, 1–9. doi: 10.1016/j.cognition.2015.02.004
- Amirshahi, A., and Hashemi, M. (2019). ECG classification algorithm based on STDP and R-STDP neural networks for real-time monitoring on ultra low-power personal wearable devices. *IEEE Trans. Biomed. Circuits Syst.* 13, 1483–1493. doi: 10.1109/TBCAS.2019.2948920
- Bendre, N., Desai, K., and Najafirad, P. (2021). Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts. *arXiv: 2106.14082*. doi: 10.1109/ICIP42928.2021.9506108
- Bichler, O., Suri, M., Querlioz, D., Vuillaume, D., DeSalvo, B., and Gamrat, C. (2012). Visual pattern extraction using energy-efficient "2-PCM synaps?? neuromorphic architecture. *IEEE Trans. Electr. Dev.* 59, 2206–2214. doi: 10.1109/TED.2012.2197951
- Bishop, C. M., Svensén, M., and Williams, C. K. (1998). GTM: the generative topographic mapping. *Neural Comput.* 10, 215–234. doi: 10.1162/089976698300017953
- Bohte, S., La Poutre, H., and Kok, J. (2002). Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer rbf networks. *IEEE Trans. Neural Netw.* 13, 426–435. doi: 10.1109/72.991428
- Braga, P. H. M., Medeiros, H. R., and Bassani, H. F. (2020). "Deep categorization with semi-supervised self-organizing maps," in *2020 International Joint Conference on Neural Networks (IJCNN)*, (Glasgow), 1–8. doi: 10.1109/IJCNN48605.2020.9206695
- Brassai, S. T. (2014). "FPGA based hardware implementation of a self-organizing map," in *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*, (Tihany), 101–104. doi: 10.1109/INES.2014.6909349
- Brownlee, A. E., Adair, J., Haraldsson, S. O., and Jabbo, J. (2021). "Exploring the accuracy-energy trade-off in machine learning," in *2021 IEEE/ACM International Workshop on Genetic Improvement (GI)*, 11–18. Madrid: IEEE. doi: 10.1109/GI52543.2021.00011
- Budayan, C., Dikmen, I., and Birgonul, M. T. (2009). Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping. *Expert Syst. Appl.* 36, 11772–11781. doi: 10.1016/j.eswa.2009.04.022
- Cain, N., Iyer, R., Koch, C., and Mihalas, S. (2016). The computational properties of a simplified cortical column model. *PLoS Comput. Biol.* 12:1005045. doi: 10.1371/journal.pcbi.1005045
- Calvert, G. A. (2001). Crossmodal processing in the human brain: insights from functional neuroimaging studies. *Cereb. Cortex* 11, 1110–1123. doi: 10.1093/cercor/11.12.1110
- Cappe, C., Rouiller, E. M., and Barone, P. (2009). Multisensory anatomical pathways. *Hear. Res.* 258, 28–36. doi: 10.1016/j.heares.2009.04.017
- Cassidy, A. S., Merolla, P., Arthur, J. V., Esser, S., Jackson, B., Alvarez-Icaza, R., et al. (2013). "Cognitive computing building block: a versatile and efficient digital neuron model for neurosynaptic cores," in *International Joint Conference on Neural Networks (IJCNN)*, (Dallas, TX). doi: 10.1109/IJCNN.2013.6707077
- Chen, C., Han, D., and Wang, J. (2020). Multimodal encoder-decoder attention networks for visual question answering. *IEEE Access* 8, 35662–35671. doi: 10.1109/ACCESS.2020.2975093
- Cholet, S., Paugam-Moisy, H., and Regis, S. (2019). "Bidirectional associative memory for multimodal fusion: a depression evaluation case study," in *2019 International Joint Conference on Neural Networks (IJCNN)*, (Budapest), 1–6. doi: 10.1109/IJCNN.2019.8852089
- Davies, M., Srinivasa, N., Lin, T., China, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359
- de Abreu de Sousa, M. A., and Del-Moral-Hernandez, E. (2017a). "Comparison of three FPGA architectures for embedded multidimensional categorization through Kohonen's self-organizing maps," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, (Baltimore, MD), 1–4. doi: 10.1109/ISCAS.2017.8050799
- de Abreu de Sousa, M. A., and Del-Moral-Hernandez, E. (2017b). "An FPGA distributed implementation model for embedded SOM with on-line learning," in *2017 International Joint Conference on Neural Networks (IJCNN)*, (Anchorage, AK), 3930–3937. doi: 10.1109/IJCNN.2017.7966351
- Diehle, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099
- Dittenbach, M., Merkl, D., and Rauber, A. (2000). "The growing hierarchical self-organizing map," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New*

- Challenges and Perspectives for the New Millennium*, Vol. 6, (Como), 15–19. doi: 10.1109/IJCNN.2000.859366
- Dong, M., Huang, X., and Xu, B. (2018). Unsupervised speech recognition through spike-timing-dependent plasticity in a convolutional spiking neural network. *PLoS ONE* 13:e204596. doi: 10.1371/journal.pone.0204596
- Douglas, R. J., and Martin, K. A. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* 27, 419–451. doi: 10.1146/annurev.neuro.27.070203.144152
- Dundar, A., Jin, J., and Culurciello, E. (2016). Convolutional clustering for unsupervised learning. *arXiv:1511.06241 [cs]*.
- Edelman, G., and Gally, J. (2001). Degeneracy and complexity in biological systems. *Proc. Natl. Acad. Sci. U.S.A.* 98, 13763–13768. doi: 10.1073/pnas.231499798
- Edelman, G., and Gally, J. (2013). Reentry: a key mechanism for integration of brain function. *Front. Integr. Neurosci.* 7:63. doi: 10.3389/fnint.2013.00063
- Edelman, G. M. (1982). “Group selection and phasic reentrant signaling: a theory of higher brain function,” in *The 4th Intensive Study Program of the Neurosciences Research Program*, (Boston, MA).
- Edelman, G. M. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection*. New York, NY: Basic Books.
- Edelman, G. M. (1993). Neural darwinism: selection and reentrant signaling in higher brain function. *Neuron* 10, 115–125. doi: 10.1016/0896-6273(93)90304-A
- Escobar-Juárez, E., Schillaci, G., Hermosillo-Valadez, J., and Lara-Guzmán, B. (2016). A self-organized internal models architecture for coding sensory-motor schemes. *Front. Robot. AI* 3:22. doi: 10.3389/frobt.2016.00022
- Fritzke, B. (1995). “A growing neural gas network learns topologies,” in *Advances in Neural Information Processing Systems*, Vol. 7. Eds G. Tesauro, D. Touretzky, and T. Leen (Denver, CO: MIT Press).
- Ghosh-Dastidar, S., and Adeli, H. (2009). “Third generation neural networks: spiking neural networks,” in *Advances in Computational Intelligence*, eds W. Yu and E. N. Sanchez (Berlin; Heidelberg: Springer Berlin Heidelberg), 167–178. doi: 10.1007/978-3-642-03156-4_17
- Girau, B., and Upegui, A. (2019). “Cellular self-organising maps-CSOM,” in *International Workshop on Self-Organizing Maps* (Denver, CO: Springer), 33–43. doi: 10.1007/978-3-030-19642-4_4
- Gorban, A., and Zinovyev, A. (2005). Elastic principal graphs and manifolds and their practical applications. *Computing* 75, 359–379. doi: 10.1007/s00607-005-0122-6
- Hazan, H., Saunders, D., Sanghavi, D. T., Siegelmann, H., and Kozma, R. (2018). “Unsupervised learning with self-organizing spiking neural networks,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, (Rio de Janeiro), 1–6. doi: 10.1109/IJCNN.2018.8489673
- Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: Wiley.
- Heylighen, F., and Gershenson, C. (2003). The meaning of self-organization in computing. *IEEE Intell. Syst.* 72–75. doi: 10.1109/MIS.2003.1217631
- Hikawa, H. (2005). FPGA implementation of self organizing map with digital phase locked loops. *Neural Netw.* 18, 514–522. doi: 10.1016/j.neunet.2005.06.012
- Indiveri, G., and Horiuchi, T. K. (2011). Frontiers in neuromorphic engineering. *Front. Neurosci.* 5:118. doi: 10.3389/fnins.2011.00118
- Jayaratne, K. M. C. (2021). *Multimodal perceptual mechanisms for unsupervised self-structuring artificial intelligence in distributed systems* (Ph.D. thesis). La Trobe Business School; College of Arts, Social Sciences and Commerce; La Trobe University, Melbourne, VIC, Australia.
- Jayaratne, M., Alahakoon, D., and de Silva, D. (2021). Unsupervised skill transfer learning for autonomous robots using distributed growing self organizing maps. *Robot. Auton. Syst.* 2021:103835. doi: 10.1016/j.robot.2021.103835
- Jin, X., Lujan, M., Plana, L. A., Davies, S., Temple, S., and Furber, S. B. (2010). Modeling spiking neural networks on spinnaker. *Comput. Sci. Eng.* 12, 91–97. doi: 10.1109/MCSE.2010.112
- Jovanović, S., Rabah, H., and Weber, S. (2018). “High performance scalable hardware SOM architecture for real-time vector quantization,” in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, (Rio de Janeiro), 256–261. doi: 10.1109/IPAS.2018.8708863
- Jovanović, S., Rabah, H., and Weber, S. (2020). “Large-scale distributed and scalable SOM-based architecture for high-dimensional data reduction,” in *AI for Emerging Verticals: Human-Robot Computing, Sensing and Networking, Computing* (Institution of Engineering and Technology), (Stevenage), 315–336. doi: 10.1049/PBPC034E_ch16
- Kamata, H., Mukuta, Y., and Harada, T. (2021). Fully spiking variational autoencoder. *arXiv preprint arXiv:2110.00375*.
- Khacef, L. (2020). *Exploration of brain-inspired computing with self-organizing neuromorphic architectures* (Thesis de doctorat). Université Côte d’Azur, Nice, France. Available online at: <https://hal.inria.fr/tel-03186924/>
- Khacef, L., Abderrahmane, N., and Miramond, B. (2018). “Confronting machine-learning with neuroscience for neuromorphic architectures design,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, (Rio de Janeiro). doi: 10.1109/IJCNN.2018.8489241
- Khacef, L., Gripon, V., and Miramond, B. (2020a). “GPU-based self-organizing maps for post-labeled few-shot unsupervised learning,” in *Neural Information Processing* (Cham: Springer International Publishing), 404–416. doi: 10.1007/978-3-030-63833-7_34
- Khacef, L., Miramond, B., Barrientos, D., and Upegui, A. (2019). “Self-organizing neurons: toward brain-inspired unsupervised learning,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, (Budapest), 1–9. doi: 10.1109/IJCNN.2019.8852098
- Khacef, L., Rodriguez, L., and Miramond, B. (2020b). Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning. *Electronics* 9:1605. doi: 10.3390/electronics9101605
- Khacef, L., Rodriguez, L., and Miramond, B. (2020c). “Improving self-organizing maps with unsupervised feature extraction,” in *Neural Information Processing* (Cham: Springer International Publishing), 474–486. doi: 10.1007/978-3-030-63833-7_40
- Khacef, L., Rodriguez, L., and Miramond, B. Written and spoken digits database for multimodal learning. *Zenodo*. doi: 10.5281/zenodo.3515935
- Kheradpisheh, S. R., and Masquelier, T. (2020). Temporal backpropagation for spiking neural networks with one spike per neuron. *Int. J. Neural Syst.* 30:2050027. doi: 10.1142/S0129065720500276
- Kim, S., Park, S., Na, B., and Yoon, S. (2020). “Spiking-yolo: spiking neural network for energy-efficient object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, (Vancouver), 11270–11277. doi: 10.1609/aaai.v34i07.6787
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69. doi: 10.1007/BF00337288
- Kohonen, T. (1990). The self-organizing map. *Proc. IEEE* 78, 1464–1480. doi: 10.1109/5.58325
- Kohonen, T. (1991). “Self-organizing maps: ophmization approaches,” in *Artificial Neural Networks* (Espoo: Elsevier), 981–990. doi: 10.1016/B978-0-444-89178-5.50003-8
- Kriegstein, K., and Giraud, A.-L. (2006). Implicit multisensory associations influence voice recognition. *PLoS Biol.* 4:e326. doi: 10.1371/journal.pbio.0040326
- Lachmair, J., MeréNyí, E., Porrmann, M., and Rüdckert, U. (2013). A reconfigurable neuroprocessor for self-organizing feature maps. *Neurocomputing* 112, 189–199. doi: 10.1016/j.neucom.2012.11.045
- Lahat, D., Adali, T., and Jutten, C. (2015). Multimodal data fusion: an overview of methods, challenges, and prospects. *Proc. IEEE* 103, 1449–1477. doi: 10.1109/JPROC.2015.2460697
- Lallem, S., and Dominey, P. F. (2013). Multi-modal convergence maps: from body schema and self-representation to mental imagery. *Adapt. Behav.* 21, 274–285. doi: 10.1177/1059712313488423
- Lang, R., and Warwick, K. (2002). “The plastic self organising map,” in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN’02 (Cat. No. 02CH37290)*, (Honolulu, HI), Vol. 1, 727–732. doi: 10.1109/IJCNN.2002.1005563
- LeCun, Y., and Cortes, C. (2010). *MNIST Handwritten Digit Database*. Available online at: <http://yann.lecun.com/exdb/mnist/>
- Lefort, M., Boniface, Y., and Girau, B. (2010). “Self-organization of neural maps using a modulated BCM rule within a multimodal architecture,” in *Brain Inspired Cognitive Systems 2010*, (Madrid), 26.
- Liu, Y., Fan, Q., Zhang, S., Dong, H., Funkhouser, T., and Yi, L. (2021). “Contrastive multimodal fusion with tupleinforce,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (Seoul), 754–763.

- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7
- Mavi, A. (2021). A new dataset and proposed convolutional neural network architecture for classification of American sign language digits. *arXiv:2011.08927*.
- Merolla, P. A., Arthur, J. V., and Alvarez-Icaza, R. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642
- Morse, A. F., De Greeff, J., Belpeame, T., and Cangelosi, A. (2010). Epigenetic robotics architecture (ERA). *IEEE Trans. Auton. Ment. Dev.* 2, 325–339. doi: 10.1109/TAMD.2010.2087020
- Muliukov, A. (2021). *Edge-Team-Leat/Scalable_Resom*. Bitbucket.
- Nakamura, T., Nagai, T., and Iwahashi, N. (2011). “BAG of multimodal LDA models for concept formation,” in *2011 IEEE International Conference on Robotics and Automation*, (Shanghai), 6233–6238. doi: 10.1109/ICRA.2011.5980324
- Orchard, G., Frady, E. P., Rubin, D. B. D., Sanborn, S., Shrestha, S. B., Sommer, F. T., et al. (2021). “Efficient neuromorphic signal processing with loihi 2,” in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, (Coimbra), 254–259. doi: 10.1109/SiPS52927.2021.00053
- Paindavoiné, M., Boisard, O., Carbon, A., Philippe, J.-M., and Brousse, O. (2015). “Neurodsp accelerator for face detection application,” in *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI, GLSVLSI '15* (New York, NY: ACM), 211–215. doi: 10.1145/2742060.2743769
- Parisi, G. I., Tani, J., Weber, C., and Wermtner, S. (2017). Emergence of multimodal action representations from neural network self-organization. *Cogn. Syst. Res.* 43, 208–221. doi: 10.1016/j.cogsys.2016.08.002
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). “Pytorch: an imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, eds H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Vancouver: Curran Associates, Inc.), 8024–8035.
- Peña, J., Vanegas, M., and Valencia, A. (2006). “Digital hardware architectures of Kohonen's self organizing feature maps with exponential neighboring function,” in *2006 IEEE International Conference on Reconfigurable Computing and FPGAs (ReConFig 2006)*, (San Luis Potosi), 1–8. doi: 10.1109/RECONF.2006.307761
- Pham, P. H., Jelaca, D., Farabet, C., Martini, B., LeCun, Y., and Culurciello, E. (2012). “Neuflow: dataflow vision processing system-on-a-chip,” in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, (Boise, ID), 1044–1047. doi: 10.1109/MWSCAS.2012.6292202
- Rathi, N., and Roy, K. (2021). STDP based unsupervised multimodal learning with cross-modal processing in spiking neural networks. *IEEE Trans. Emerg. Top. Comput. Intell.* 5, 143–153. doi: 10.1109/TETCI.2018.2872014
- Rodriguez, L., Khacef, L., and Miramond, B. (2018). “A distributed cellular approach of large scale SOM models for hardware implementation,” in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, (Athens), 250–255. doi: 10.1109/IPAS.2018.8708904
- Rougier, N., and Boniface, Y. (2011). Dynamic self-organising map. *Neurocomputing* 74, 1840–1847. doi: 10.1016/j.neucom.2010.06.034
- Schliebs, S., and Kasabov, N. (2013). Evolving spiking neural networks—a survey. *Evol. Syst.* 4, 87–98. doi: 10.1007/s12530-013-9074-9
- Schuman, C. D., Disney, A., and Reynolds, J. (2015). “Dynamic adaptive neural network arrays: a neuromorphic architecture,” in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, MLHPC '15*, (Austin, TX), doi: 10.1145/2834892.2834895
- Shatz, C. J. (1992). How are specific connections formed between thalamus and cortex? *Curr. Opin. Neurobiol.* 2, 78–82. doi: 10.1016/0959-4388(92)90166-1
- Singer, W. (1990). The formation of cooperative cell assemblies in the visual cortex. *Journal of Experimental Biology*, 153, 177–197. doi: 10.1242/jeb.153.1.177
- Smith, L. and Gasser, M. (2005). The development of embodied cognition: six lessons from babies. *Artif. Life* 11, 13–29. doi: 10.1162/1064546053278973
- Upegui, A., Girau, B., Rougier, N., Vannel, F., and Miramond, B. (2018). “Pruning self-organizing maps for cellular hardware architectures,” in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, (Edinburgh), 272–279. doi: 10.1109/AHS.2018.8541465
- Vannel, F., Barrientos, D., Schmidt, J., Abegg, C., Buhlmann, D., and Upegui, A. (2018). “Scalp: self-configurable 3-d cellular adaptive platform,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, (Bangalore), 1307–1312. doi: 10.1109/SSCI.2018.8628794
- Varela, F. J., Rosch, E., and Thompson, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge, MA: MIT Press. doi: 10.7551/mitpress/6730.001.0001
- Vasco, M., Melo, F. S., and Paiva, A. (2020). MHVAE: a human-inspired deep hierarchical generative model for multimodal representation learning. *arXiv preprint arXiv:2006.02991*.
- Voegtlin, T. (2002). Recursive self-organizing maps. *Neural Netw.* 15, 979–991. doi: 10.1016/S0893-6080(02)00072-2
- Warden P. (2018). Speech commands: a dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*. Available online at: <https://arxiv.org/pdf/1804.03209.pdf> (accessed April 09, 2018).
- Werker, J., Cohen, L., Lloyd, V., Casasola, M., and Stager, C. (1998). Acquisition of word/object associations by 14-month-old infants. *Dev. Psychol.* 34, 1289–309. doi: 10.1037/0012-1649.34.6.1289
- Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*.
- Xie, J., Zhu, Y., Zhang, Z., Peng, J., Yi, J., Hu, Y., et al. (2020). “A multimodal variational encoder-decoder framework for micro-video popularity prediction,” in *Proceedings of The Web Conference 2020, WWW '20* (Taiwan: Association for Computing Machinery), 2542–2548. doi: 10.1145/3366423.3380004
- Xin, J., and Embrechts, M. (2001). “Supervised learning with spiking neural networks,” in *IJCNN'01, International Joint Conference on Neural Networks, Proceedings (Cat. No.01CH37222)*, (Washington, DC), Vol. 3, 1772–1777.
- Younis, B. M., Mahmood, B., and Ali, F. H. (2009). Reconfigurable self-organizing neural network design and its FPGA implementation. *AL Rafdain Eng. J.* 17, 99–115. doi: 10.33899/rengj.2009.42925
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., et al. (2016). Apache spark: a unified engine for big data processing. *Commun. ACM* 59, 56–65. doi: 10.1145/2934664

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Muliukov, Rodriguez, Miramond, Khacef, Schmidt, Berthet and Upegui. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.