Columnar Learning Networks for Multi-Sensory Spatiotemporal Learning

Sangmin Yoo, Yongmo Park, Ziyu Wang, Yuting Wu, Saaketh Medepalli, Wesley Thio and Wei D. Lu*

S. Yoo, Y. Park, Z. Wang, Y. Wu, S. Medepalli, W. Thio, Prof. W. D. Lu

Department of Electrical Engineering and Computer Science

The University of Michigan

Ann Arbor, MI 48109, USA

*E-mail: wluee@umich.edu

-Abstract

Network features found in the brain may help implement more efficient and robust neural networks. Spiking networks process spikes in the spatiotemporal domain and can offer better energy-efficiency than deep neural networks. However, most SNN implementations rely on simple point neurons that neglect the rich neuronal and dendritic dynamics. In this study, we propose a bio-inspired columnar learning network (CLN) structure that employs feedforward, lateral and feedback connections to make robust classification with sparse data. The network is inspired by the mammalian neocortex, comprising cortical columns each containing multiple minicolumns formed by interacting pyramidal neurons. A column continuously processes spatiotemporal signals from its sensor, while learning spatial and temporal correlations between features in different regions of an object along with the sensor's movement through sensorimotor interaction. The columnar learning network can be implemented using memristor crossbars with a local learning rule, Spiking Timing Dependent Plasticity, that can be natively obtained in 2nd-order memristors. CLN allows inputs from multiple sensors to be simultaneously processed by different columns, resulting in higher classification accuracy and better noise tolerance. Our analysis on networks implemented on memristor crossbars show the system can operate at very low power and high throughput, with high accuracy and robustness to noise.

1. Introduction

Neural networks have recently attracted much attention for perception and learning tasks.[1,2] Despite these advances, common deep neural networks (DNNs) lack biological features that can potentially make the system more efficient and robust. Spiking neural networks (SNN) use spikes and temporal encoding methods to process information and are potentially more efficient than artificial DNNs.[3–9] Bio-inspired neuromorphic systems are also compatible with emerging devices such as memristor crossbars that allow efficient hardware implementation[10–18]. However, most SNN implementations use point neurons that neglect the rich internal dendritic structures of cortical neurons and the spatiotemporal computing capabilities of the dendrites. In this study, we propose a Columnar Learning Network (CLN) that utilizes these different dendritic structures for efficient multi-sensory, spatiotemporal data processing. The CLN is inspired by the theory that the neocortex is comprised of cortical columns, each of which consists of identical components - pyramidal neurons and synaptic connections trained by local learning rules.[19–22] Key components of the CLN include feedforward connections from the input, lateral connections that allow firing neurons to predict neighboring neurons' action potentials at the next time step, feedback connections from firing neurons at the next layer, and inter-column connections that enable columns to collectively perform a higher order intelligent task than what a single column can do. It is different from several prior studies that have aimed to adopt the concept of pyramidal neurons to memristor-based SNN, but with a focus at the neuron level rather than the construction of a bio-plausible network with cortical column structures as building blocks.[23]

The proposed CLN can be implemented efficiently with 2nd-order memristors that natively possess the local learning rule, spiking time dependent plasticity (STDP)[24–26]. We test the network with different inputs (e.g. visual and auditory), and verify its performance in terms of accuracy and noise robustness in object recognition tasks. With the columnar structure, each column can receive inputs from different sensors that detect the same object, and the multiple columns work collectively to produce more robust classification results. We further estimate the performance of the hardware-implemented network and show the proposed network can achieve extremely low energy consumption and high throughput.

2. Main

2. 1. Column structures

The neocortex is organized horizontally into six laminae, and vertically into columns of cells linked via synaptic connections across laminae. The basic unit of the neocortex is the minicolumn which is a narrow chain of neurons extending vertically,[19] and multiple minicolumns form a cortical column by short-range horizontal connections. Many columns exist in the neocortex, and each column may take inputs from a different sensor. **Figure 1**a schematically shows the column concept.

In vision, the input a column of neurons processes forms a receptive field (RF), as illustrated by the colored circles in Figure 1a. Cells in one minicolumn process a local region (e.g. marked by the yellow circles) in the RF, and different minicolumns may process different local regions in the RF.[27] Neuron firing in a

minicolumn is typically sparse, [28] e.g. following behaviors such as winner-take-all (WTA) that at most one cell in a minicolumn can fire per input and others are inhibited by the winner.[21,28] Through this competition, each cell in a minicolumn may detect a different input pattern. While the minicolumns execute relatively simple feature detection, synaptic connections along or across columns allow the network to execute intelligent tasks of higher degree.[29]

2. 1. 1. Pyramidal neuron model

The minicolumns are formed by pyramidal neurons[19,30]. Figure 1b schematically shows a pyramidal neuron structure, consisting of a soma, an axon, and dendrites that are further categorized into basal and apical dendrites. The basal dendrites (BD) descend from the base of the soma, while the apical dendrites (AD) ascend from the top of the soma.[31] Functionally, the basal dendrites process feedforward signals from the lower layer as well as lateral signals from firing neurons within or across the minicolumns[32–35]; whereas the apical dendrites processes feedback signals from firing neurons in the upper layer.[20,35–37] The basal dendrites are further divided into proximal basal dendrites (PBD) and distal basal dendrites (DBD) depending on their adjacency to the soma.[38] The PBD are closer to the soma and due to their proximity produce strong signals and have a direct impact on the soma's firing, while the far away DBD inputs are weaker due to attenuation along the dendrites, and are incapable of directly inducing neuron firing.[39] However, spikes from the DBDs can still modify the neuron membrane potential and increase the probability of the neuron winning within a minicolumn. Similar to the effects of DBDs, feedback signals collected by apical dendrites indirectly affect neuron firing.

2. 1. 2. CLN structure

Figure 1c schematically shows an example of the proposed CLN. In this example, there are 2 cortical columns, each of which has 5 minicolumns at each horizontal layer, where the horizontal layers correspond to the laminae layers in Figure 1a. The two columns can receive inputs from two different sensors, e.g. visual and audio inputs, respectively. Each minicolumn in turn has 4 pyramidal neuron cells. As discussed earlier, cells in the same minicolumn receive identical inputs.

In the proposed CLN, neurons in Layer 1 receive inputs through their PBDs. Spikes generated by the winning neurons (marked in green) in Layer 1 are propagated onto Layer 2 through axonal connections (depicted as orange arrows) and reach cells in Layer 2 via the PBDs of the Layer 2 neurons. In this example, the winning cell (marked in blue) in Layer 2 produces the classification output, and in turn sends backpropagating spikes to cells in Layer 1 through their ADs (marked in purple).[20]

Beyond feedforward (via PBDs) and feedback connections (via ADs), spikes from the winning neurons also propagate to other cells in the same horizontal layer (both within a minicolumn and across minicolumns) through the DBDs of those cells, as illustrated by the red arrows in Figure 1c. These lateral connections allow the firing neurons to send signals to "predict" the action of their neighboring neurons in the next step.

To illustrate how these different connections, i.e. feedforward, lateral and feedback connections, work, let's examine a case where the two green cells in Layer 1 of column 1 fire at time t_1, as shown in Figure 1c. These spikes will be propagated to other neurons in Layer 1 via the lateral DBD connections (red arrows). The firing cells in Layer 1 also cause a cell (blue) in Layer 2 to fire, and feedback is sent through the ADs to all cells in Layer 1, as shown in the Figure 1c.

At the next time t_2, cells in Layer 1 receive new inputs through the feedforward connections via the PBDs. The effects of the feedforward signals are then combined with the prediction signals from the lateral DBDs and the feedback signals from the ADs. Note the prediction signals and the feedback signals are from spikes generated at a prior time t_1. The effects of the different signals also depend on the connection strength, with PBDs typically having the strongest connections and DBDs and ADs having weaker connections. The collective effects of the feedforward, lateral and feedback signals produce the new winning neuron at time t_2, marked in yellow. This process is then repeated.

We expect the operations of the CLN will be more efficient and robust when compared with feedforward-only networks, because the lateral and feedback signals enable spatiotemporal cues among neurons. For example, when an input induces comparable membrane potential changes in multiple cells in one minicolumn, the winner can still be determined due to the lateral and feedback spikes from other neurons that fired based on earlier inputs. In turn, these connections can make the CLN effective in learning spatiotemporal patterns in the input and make robust decisions.

2. 1. 3. Sensorimotor integration approach

We adopt the sensorimotor integration approach to send inputs to the cortical columns over time, and to enable the network to process time-series information in the form of data streams.[21,40] Briefly, visual sensors generate spikes from a localized region corresponding to the RF, where the eyes are staring at, and new signals are continuously generated as the RF is updated along with the eyes' movement over time (Figure 1e). It is a common workflow for humans to classify an object too large to be captured at once, and instead interprets the spatial information of an object in the temporal domain.

Once the vision sensor captures a portion of the object and sends the visual signals, the brain extracts features from the data and predicts candidates of what the object may be, using the extracted features and the location information of the focal points. Afterwards, it outputs motor signals to the sensor's muscle to shift the focal points to another region to get new inputs. The process is depicted in Figure 1d,e. Through the repeated cycles (Figure 1d), the brain selects the possible candidates of the object and determines what it is in the end. During this process, the brain may also learn correlations between features found in different regions of the object and the sensor's movements, allowing it to predict more accurately in the next task using the learned correlation.[41,42]

The coordinated sensor-motor actions make the system very efficient in object detection once the correlation has been learned. Figure 1f shows an example of finding a constellation in the sky. Following the

learned correlation, the eyes will preferentially focus on the regions where the stars belonging to the constellation may exist, marked as the shaded regions in Figure 1f. This process eliminates the need to produce and process inputs at other regions not related to the constellation, thus greatly reducing the total energy cost.

2. 1. 4. Classification Task Example

Figure 1g-i illustrate an example of the CLN network performing classification task using the common visual dataset, MNIST. For this single sensor (input stream) task, only one cortical column is needed. Following the sensorimotor integration approach, Layer 1 of the column receives inputs from a portion of an image that corresponds to its RF at a given timestep and scans the image over time. For convenience, we choose the direction of the sensor scan to be the vertical direction, and an RF area corresponds to a horizontal stripe of the image. No inputs are provided at time t_1, and the green stripes labeled t_2 to t_{14} correspond to the RFs of the visual inputs to the CLN during the sensor navigation, with some overlap between the RFs at consecutive timesteps.

At time t_2, input spikes are sent to neurons in Layer 1 of the CLN, based on the intensity of the pixels. Each minicolumn in Layer 1 processes a specific region in the RF (for example, those marked by the blue rectangles in Figure 1g) causing some cells to fire. The firing cells in turn propagate spikes to other cells in the same minicolumn and across minicolumns through DBDs, which modulate the likelihood of these cells to fire at the next timestep. During this process, the network learns temporal tendency of the input features through the DBD weight updates.

As shown in Figure 1c,e, the range of DBDs is spatially short so that the prediction effect from firing neurons is mainly to its neighbors. The spatial range of the prediction at the next timestep is illustrated as green pyramids in Figure 1h. The firings of cells in Layer 1 additionally cause certain cells in Layer 2 that represent possible candidates (numbers 0, 8 and 9 in this example) to fire. These spikes in turn provide feedback signals to cells in the lower layer through ADs that affect which cells in Layer 1 may fire in the future, along with the input spikes and the lateral prediction spikes. This process is repeated throughout the sensor movement from the top to the bottom. In the end, the cell in Layer 2 that fires the most times during the scan is chosen in the end as the result of the classification. This example is schematically shown in Figure 1i for the three candidate cells in Layer 2, with red and green color representing silence and firing of the cell during sensor movement, respectively. In the end, the cell corresponding to number 8 is chosen as the classification result since it produced the most spikes over time.

2.2. CLN implementations

2.2.1 Basal dendritic connections

A CLN composed of a single cortical column across two layers is illustrated in **Figure 2**a, highlighting the PBDs, DBDs, axons and ADs for the pyramidal neurons.

In this example, we use the single column CLN to process inputs from a single sensor, (e.g. the MNIST

dataset). As explained above, at any given time an RF corresponding to a 32x3 stripe in the input is processed by the Layer 1 neurons through the PBD connections, as shown in Figure 1g,h. We chose conventional feedforward networks to model the PBD connections. Specifically, the PBD connections are modeled by a 2x2 convolution layer with depth of 4, followed by a 2x2 summation pooling layer, as shown in Figure 2c. The depth here represents the number of cells in a minicolumn, as all cells in a minicolumn receive identical inputs but can learn different features. The 2x2 pooling layer improves the network's robustness against minor spatial or temporal displacement of input signals.

After 2x2x4 convolution and 2x2 pooling in the PBD connections, the original 32x3 input is converted into a 15x1x4 feature map that corresponds to the input-induced changes of the soma's membrane potential, across15 minicolumns with 4 neurons in each minicolumn. To implement WTA in a minicolumn, we adopted a depth-wise inhibition (DWI) operation which allows at most one of the four cells per minicolumn to fire.

Following the Sensorimotor integration approach, after the first RF is processed, the sensor moves to the next stripe along the vertical direction. We allow an overlap of 1 row of pixels between the RFs, so the original 32x32 data correspond to 15 RFs, each corresponding to a 32x3 stripe with an overlap of 1 row. The RFs are processed by the CLN sequentially over time t_1 to t_{15}. In this example (Figure 1g), there are no data in the first and last stripe, so the CLN only need to operate during t_2 to t_{14}.

2.2.2 Distal dendritic connections

The output spikes from firing neurons in Layer 1 are then sent to other neurons in Layer 1 through the lateral DBD connections, as depicted in Figure 2a. We model the DBD connections with small fully connected (FC) networks, where neurons in one minicolumn are connected to other neurons in r neighboring minicolumns (including itself), as shown in Figure 2b,d. In this example, the DBD connections for neurons in one minicolumn are represented by a 4x4r network, and different minicolumns will have different DBD connections trained using unsupervised local learning such as STDP.

2.2.3 Output layer and Apical dendrites

Spikes from neurons in Layer 1 are also sent to neurons in Layer 2 via axonal connections for further processing, e.g. classification. These axonal connections are modeled as feedforward connections and implemented using FC layers. Specifically, there are 60 neurons in Layer 1 generating data over 15 timesteps, and 10 neurons in Layer 2 corresponding to the 10 output classes in this example. The simplest approach is to connect all outputs (60x15) from Layer 1 with all neurons in Layer 2, using a 900x10 FC layer. However, this will result in significant hardware overhead. Again, following the sensorimotor integration approach and the fact that Layer 1 neurons only process local spatiotemporal data, we can split the 15 timesteps into groups that focus on local temporal features. For example, the 15 timesteps can be split into 5 groups, each corresponding to 3 consecutive timesteps. Consecutive neuron outputs during the 3 timesteps are used as the input to Layer 2 neurons. We note this approach is similar to the temporal accumulation effect employed in Reservoir Computing systems[43,44] to detect the temporal patterns over time. In theory, the consecutive outputs can also

be replaced by a trainable 3x1 convolution, but our studies show there is little difference in the final classification accuracy between using a trained 3x1 convolution and the simple outputs from the 3 consecutive timesteps.

Using this approach, the 900x10 FC layer can then be replaced with 5 60x10 FC layers, as shown in Figure 2e. The spiking cells in Layer 2 correspond to classification outputs, and these spikes are in turn sent back to Layer 1 neurons through the feedback AD connections. In our design, to save hardware and training costs, we use the same 5 60x10 axonal connections for the AD connections, with signals running backwards across the network when used as AD connections. As discussed earlier, the feedback signals are combined with the lateral DBD signals and the new input signals to determine the Layer 1 neuron firing in the next time step.

2.2.4. Multi-column networks

A key feature of the CLN is its ability to use multiple columns to execute tasks in parallel or execute higher order intelligent tasks, where each column receives input from a different sensor or distributed inputs (e.g. from the tactile system), following the principles shown in Figure 1c. Figure 2f shows an example of a two-column network, where each column is configured in the exact same manner, and all hyperparameters are identical in the columns.

Possible examples of dual-sensor inputs are right eye and left eye, eyes and ears, or other pairs of sensory organs. For the demonstration, we selected artificial visual and audio inputs based on the MNIST and FSDD datasets, respectively, to mimic cortical columns receiving signals from eyes and ears.[45,46] We preprocessed both datasets to make them the same dimension of 32x32. Like the single column CLN case, each sensor takes a 32x3 region from the input per timestep.

In the training phase, each column is trained individually using training data in its respective dataset. After training, each column can analyze input data received by its own sensor. Additionally, decision is now derived by the aggregate of the cell potentials from the corresponding Layer 2 cells in both columns. The aggregation is executed by the inter-column connections as depicted by the yellow arrow in Figure 2f, where the membrane potentials of corresponding Layer 2 neurons from the 2 columns are summed up in an element-wise fashion. During inference tests, we rearranged the two separate test datasets into a combined test dataset, grouped by common ground truths. Due to the smaller audio dataset, audio data is re-used multiple times in the combined test dataset to resolve the unbalanced number of available test data.

2.2.5. Preprocessing

Before feeding data to the network, we preprocessed the raw data. First, analog values of the visual inputs and the spectrogram of audio inputs are normalized. Second, we binarized and then skeletonized the visual inputs, while audio inputs are whitened and then binarized. **Figure 3**a,b show an example of the original visual and audio inputs representing the same object, and Figure 3c,d show results after the preprocessing step for both cases. For the auditory data, the x-axis represents frequency, and the y-axis represents time (from top

to bottom).

2.3. CLN Training

2.3.1 DBD connections

The DBD connections are trained using local learning rules in an unsupervised manner. Specifically, we leverage the STDP effects that can be natively implemented in a 2nd-order memristor with simple, non-overlapping spikes. [24–26] Among our fabricated 2nd-order memristor devices, we chose the type with the shortest internal time constants to decrease the operation latency of the system. The experimentally obtained STDP characteristics for such a device are shown in Figure 3i, along with fitting curves. The conductance modulation curves when subjected to a series of pre- and post-pulse pairs are shown in Figure 3j.

Following [47], the weight update dynamics by a pair of pre- and post-spikes can be described as:

$$\Delta W = W_A \cdot exp(-\frac{|\Delta t|}{\tau}) \cdot (W_{LRS} - W) \cdot (W - W_{HRS}) \tag{1}$$

where W_A is a fitting parameter, Δt is the time gap between the pre and post spikes, τ is the STDP time constant, and W is the synaptic strength. W_A and τ may be different for LTP and LTD.

The DBD connections for neurons in one minicolumn are implemented in a memristor crossbar, as shown in Figure 3k, where each connection is implemented with a 2nd-order memristor. Inputs to the crossbar are neuron firings from the minicolumn, which modulate the membrane potential of neurons in the neighboring r minicolumns (itself included) based on the input spikes and the synaptic weights. The input and output neuron spikes in turn natively update the corresponding synaptic weights, through the STDP effect described in Equation 1.

2.3.2 PBD connections

The PBD connections detect local features in the input. Instead of training the PBD connections, we chose to use a pre-selected 2x2x4 convolution kernel, shown in Figure 3l, for the convolution layer in the PBD to minimize hardware and training cost. These pre-selected features detect different slopes in the spatiotemporal domain (e.g. angles in visual inputs, and shifts in frequency-time in audio inputs). Figure 3e,f show example outputs after convolution with the 4th feature in the 2x2x4 convolution kernel, for the visual and audio inputs, respectively. Figure 3g,h show outputs of the winning neurons in Layer 1 after integrating contributions from the PBD connections that consist of 2x2x4 convolution, 2x2 summation pooling, and DWI, as well as contributions from the DBD connections and the AD connections.

2.3.3 Axonal connections and AD connections

The axonal connections and AD connections are also implemented in memristor crossbars using the same 2nd-order memristor devices as used in the DBD connections. For the axonal connections, 5 60x10 crossbars are used. Since Layer 2 neurons need to perform classification, the weights are trained using a supervised learning algorithm, softmax regression, using accumulated spikes from the 10 Layer 2 neurons that

correspond to the 10 output classes. The cost function is calculated following the negative log likelihood loss function (NLLLoss) method and minimized by a standard gradient-based optimization method, RMSprop, through weight updates.[43,44,48] The same trained axonal connections are also used as the AD connections for the feedback signals, where spikes from the Layer 2 neurons are propagated backwards through the crossbars.

2. 4. Results

We verified the performance of the implemented CLN network in terms of classifying accuracy, noise robustness, and spike sparsity using neural network simulations. During the verification stage, we also analyzed how the neighborhood range, r, of the DBD connections and the number of groups in the axonal connections affect the network performance.

2.4.1 CLN performance

For each CLN design, we ran 30 trials of training and testing, and averaged the results of the trials to minimize effects arising from random weight initialization at the beginning of the training process. It roughly takes 15 epochs to achieve saturated classifying accuracy for both visual and audio datasets, as shown in **Figure 4**a,b. Very little accuracy drop is observed during testing for the visual inputs (Figure 4a). For the auditory inputs, a 1.5% loss in test accuracy is observed due to the small auditory dataset (2700 training samples vs 60000 training samples in the visual dataset) that make the training less effective, even after adopting techniques such as dropout during training (Figure 4b).

More interestingly, Figure 4c shows that the multicolumn network introduced in Figure 2f can achieve meaningfully higher accuracy in the classification task than both single column networks. Even though each column is trained separately in the two-column CLN, the inter-column connections help the network make more robust decisions. This effect may be traced back to the network's biological inspirations, where it is common that a person is better at classifying an object when he or she is using multiple sensors like two hands, or hand and eyes, or eyes and ears, rather than using only a single sensor. To verify the higher accuracy is due to the cooperative effects of both sensor inputs instead of simply the larger network size, we compared the two-column CLN results with single-column networks that are double the original size, either by making the network wider (by using wider dimension feature maps) or deeper (by additionally adding more layers), as shown in Figure 4c and Table 1. The two-column network still outperforms these larger single column networks that use only visual data, suggesting the multicolumn network is more efficient by leveraging inputs and features from multiple sensors.

2.4.2. Effects of network structure

Figure 4d shows how the DBD connection's neighborhood range r affects the network accuracy. It is interesting that after initially improving the accuracy, further increasing the range leads to accuracy drop. It is possible that predictions made far away from a neuron may lack spatiotemporal correlation and will just add

noise. This observation is consistent with the theory that attenuation along the dendrites plays an important role, and effects from far away neurons are diminished considerably[49]. In the following discussion, we reduce the DBD neighborhood range from 5 to 3 to achieve better power and area efficiency with minimal accuracy drop. The corresponding numbers of memristors used to implement the DBD connections for different *r* are shown in Table 2.

Next, we tested the effects of grouping in the axonal connections between Layer 1 and Layer 2 neurons. As discussed earlier, without grouping (group size = 1), a large 900x10 FC layer is required. Grouping outputs from 3 consecutive timesteps reduces the size of the axonal connections to 5 60x10 FC layers with minimal accuracy change (Figure 4e). Further increasing the group size (e.g. to 5) leads to a more measurable accuracy drop with reduced benefits in network size reduction, as shown in Table 3. To balance the performance and the hardware cost, we set the group size to 3 in the following discussions.

Feedback signals through the ADs help the network to achieve better accuracy in classification tasks and improve its noise tolerance. Figure 4f shows the effects of ADs on the auditory task in a single-column CLN, where clear disparity in terms of accuracy can be observed with and without the AD connections.

2.4.3. Noise tolerance

A key potential advantage of the CLN over conventional DNN is better noise tolerance provided by the lateral and feedback connections. We tested two types of noise that are likely to both sensory inputs - insufficient data and Gaussian noise. The former is by providing only a portion of the data, and the latter is by adding Gaussian noise with mean 0 and different variance to the input data.

For comparison with widely known networks, we chose LeNET which is one of the smaller DNNs but still ~8x the size of the two-column CLN to benchmark the noise performance.[50] Figure 4g,h show how well the networks deal with insufficient data and Gaussian noise, respectively. In both cases, the proposed CLN network outperforms LeNET, and the robustness to noise is highly apparent at higher noise levels in both cases. In addition, the multi-column network also shows superior noise tolerance to single columnar networks, supporting the hypothesis that utilizing multiple sensory inputs leads to more robust decisions.

2. 4. 4. Spike sparsity

The proposed CLN network produces sparse spikes due to competition of neurons within a minicolumn and the inhibitory effects from the lateral and feedback connections. Figure 4i shows the total number of spikes generated by all cells in Layer 1 for the 2 tasks, at different network configurations. Increasingly sparser spike counts are obtained with the addition of WTA implementations, lateral connections, and feedback connections, respectively, potentially making the network more energy efficient. The spike counts of these different configurations are listed in Table 4.

2. 5. Hardware implementation

2. 5. 1. Hardware architecture

We next study the hardware implementation of the CLN to estimate energy consumption and speed of the proposed network. For better parallelism and reconfigurability of the network, we mapped the network in a tiled-architecture.[51,52] For the very small sized CLN discussed here, we chose 4x4 and 12x12 crossbars as the fundamental building blocks, although larger crossbars are likely needed for more complex CLN designs. **Figure 5**a shows the allocation of the crossbar arrays to each layer for a single CLN column, following the network structure shown in Figure 2a. Blue, yellow, and green blocks represent PBD, DBD and axonal/AD connections, respectively. The PBD weights are copied three times to increase parallel processing of the original input data. The crossbar arrays used to map different connections are summarized in Table 5. Detailed description of the hardware implemented CLN can be found in Methods.

Figure 5b shows the implementation of the PBD connections in detail, including the 2x2x4 convolution kernel weight matrix (mapped in a 4x4 crossbar, a blue block), 4 integrators and comparators (brown blocks) to implement threshold functions that binarize the induced potential changes, and 4 integrate and fire (IF) neuron circuits (violet squares) for the neurons in a minicolumn, and a MAX circuit that finds the largest membrane potential among the 4 neurons for DWI operation. The neuron with the highest potential will fire if its potential also crosses a pre-determined threshold.

The DBD connections comprise 5 12x12 crossbars, with one such example shown in Figure 5c. For a neighborhood size of 3, the 12 crossbar columns are connected to the neurons in 3 minicolumns (MC_X, MC_Y, MC_Z). With 3 copies of PDB connections, 3 inputs can be processed in parallel and 3 MCs (MC_a, MC_b, MC_c) may spike at a given time.

The 5 60x10 axonal connections are mapped in 25 12x12 crossbars, as shown in Figure 5d, where the 5 groups during sensorimotor movement are illustrated in different shades of green. Spikes from the 10 output neurons in each group correspond to classifications made during the sensorimotor movement, and the accumulated spikes over the groups produce the final classification output.

2. 5. 2. Performance estimation

Next, we estimate the energy consumption and speed of the network for processing the different tasks based on the forementioned hardware. We used the TSMC 28nm technology library to synthesize the required circuits to obtain realistic power and speed estimates.[53,54] We set pulse amplitude and width of a spike to 0.3V and 10ns to prevent unexpected weight updates during inference.

The energy consumed by a crossbar array in the PBD, DBD and axon/AD connections is estimated as:

$$E_{\mathrm{array}} = V_{\mathrm{spk}} \cdot t_{\mathrm{spk}} \cdot \sum_{t=1}^{T} \sum_{x=1}^{X} S(x,t) \cdot W_{\mathrm{array}} \tag{2}$$

where V_{spk} and t_{spk} are the pulse amplitude and width of a spike, $S(x,t)$ is a delta function that equals to 1 if there is a spike at position x and time t in the input and 0 otherwise, and W_{array} is the weight matrix of the layer.

The energy consumed by the threshold function used in the PBD connections is estimated as:

$$E_{TH} = E_{\text{comp}} \; x \; \sum_{t=1}^{T} \sum_{x=1}^{X} \sum_{d=1}^{D} B_{\text{CONV}}(d,x,t) \tag{3}$$

where *T*, *X* and *D* are the *y* (here representing time), *x* (representing position) and depth dimension of the output feature map of the convolution layer, E_{comp} is the energy required by the comparator per cycle, and $B_{\text{CONV}}(d,x,t)$ is the binarized membrane potential for position *x* in depth *d* at time *t*.

The energy consumed by the DWI function is estimated as:

$$E_{DWI} = \sum_{t=1}^{15} \sum_{x=1}^{15} E_{\text{MAX}} + \sum_{t=1}^{15} \sum_{x=1}^{15} \sum_{d=1}^{4} TH(V_{\text{MAX}}, V_{\text{mem}}(d,x,t)) \cdot E_{\text{comp}} \tag{4}$$

where E_{MAX} is the energy consumed by the MAX circuit, *TH(a,b)* is a threshold function that outputs 1 when *b* is same or larger than *a*, V_{MAX} is the largest potential found by the circuit, $V_{\text{mem}}(d,x,t)$ is the membrane potential of cell *d* in minicolumn *x* at time *t*, and E_{comp} is the energy of the comparator which also appears in Equation 3.

Using Equation 2-4, we simulated the total energy the CLN spends to process each input. For the calculation, we used the test dataset of both sensory inputs and averaged the energy per input over the whole dataset. The values are presented in Table 6. The higher energy consumption for the auditory inputs is explained by the fact that the visual inputs have sparser spikes than the auditory inputs, as illustrated in Figure 3c,d, Figure 4i and Table 4.

The latency of the system is calculated by counting the number of timesteps to process an input data, assuming each timestep is 2x the width of a spike's width (10ns). As discussed earlier, since the convolution operations in the feedforward connections consume the most timesteps, we employed three copies of the PBD connections to speed up the process.

The estimated energy and latency per input, as well as the overall system power, are summarized in Table 7. The small network size, combined with the sparsity of spikes, make the proposed network extremely energy efficient compared with other SNN networks reported in literature based on similar technology[55]. Overall, the network can operate at < 0.5mW and is capable of processing inputs at very high inference rates (e.g. thousands of inferences per second, IPS) with high accuracy and noise robustness, as discussed earlier.

3. Conclusion

In this study, we proposed a bio-inspired network structure that can be implemented with memristor crossbars for efficient and robust processing of multi-sensory, spatiotemporal data. The network is inspired by the cortical column structure in the mammalian neocortex. We show that lateral and feedback connections make the network more efficient and robust, and the inter-column connection allows the multi-column network to achieve higher accuracy and greater noise robustness compared to single sensor networks.

The CLN network can be efficiently mapped on memristor crossbars. The local STDP rule used in the lateral connections in CLN can also be natively implemented with the internal dynamics of a 2$^{\text{nd}}$-order

memristor. Our simulation of the proposed network shows extremely low energy consumption and high throughput, suggesting the potential of the proposed network for low-energy, real-time on-sensor applications.

4. Methods

4. 1. Device fabrication

The TaO_x based 2nd-order memristors are fabricated following prior studies.[25,26] Device fabrication starts with 35nm Pd bottom electrode deposition by photolithography, e-beam evaporation and lift-off, followed by sputtering of 30nm TaO_x using a Ta metal target in an Ar/O_2 gas mixture (3% O_2, ~5mTorr) at 400°C. A 4-nm Ta_2O_5 switching layer is then deposited by RF sputtering using a Ta_2O_5 ceramic target in Ar at ~5 mTorr. The 40nm Pd top electrode is fabricated and deposited in the same way as the bottom electrode.

4. 2. Device Measurement

The 2nd-order memristors were measured with pulse pairs to extract the STDP behavior and weight update curves. For STDP measurements (Figure 3i), we applied the pre-synaptic pulse with a 0.75V amplitude and 100ns width to the top electrode, and post-synaptic pulse with a 0.7V amplitude and 100ns width to the bottom electrode. The time gap between the pulses is adjusted from -300ns to 300ns. For the weight update curves (Figure 3j), we applied a fixed number of such pulse pairs consecutively with a fixed time gap of 150ns.

4. 3. Preprocessing of the datasets

For visual inputs, we normalized and binarized the analog pixel values of the written digits in the MNIST dataset[45] using a prefixed threshold value of 0.6. Afterwards, the data are skeletonized using the scikit-learn skeletonize function[56]. For audio inputs, we normalized the analog values in the FSDD[46] dataset and whitened the data with epsilon of 0.4.[57] After that, we applied 2x2 maxpooling to the FSDD data to change the input dimension to 32x32, i.e. the same as the inputs in the MNIST dataset. Afterwards, we binarized the inputs using a prefixed threshold value of 0.2.

4. 4. Power and latency estimate of CLN hardware

To estimate the hardware power and latency, we synthesized necessary peripheral circuits including comparators, IF neuron circuits and MAX circuits (which find the largest membrane potential), following the literature.[53,54] We counted the number of timesteps and the number of spikes generated during the Python simulation. Using the information, we calculated the energy consumption and the latency required to complete the tasks. We ignored some parasitic factors like wire resistance and parasitic capacitance in these estimates.

The weight matrix of the convolution layer and the comparators in the PBD connections consume energy every cycle. The IF neuron circuits and the MAX circuit in the DWI circuit work every four cycles to find a winner within a minicolumn. This process is repeated for the 15 minicolumns for each 32x3 receptive field.

When a winner of the minicolumn fires, the output spike is propagated through a DBD weight array as shown in Figure 5c. When three copies of the convolution kernels are used to reduce latency, 3 minicolumns may output spikes to the same neighborhood at a given time. In this case, the active minicolumns are dynamically assigned to the input of the FC layers to maximize hardware utilization.

The spikes are also propagated to the axonal connections, shown in Figure 5d. The 5 groups during the sensorimotor movement are shown as 5 different shades of green. Within each group, i.e. during time t_1-t_3, the 10 output neurons will fire based on the accumulated outputs from the crossbars. The neuron spiking corresponds to classifications made during the sensorimotor movement. Spikes from the corresponding neurons are then accumulated over the 5 groups and produce the final classification for the object.

Table 1. Network structures of the two-column CLN and the wider and deeper CLNs shown in Figure 4c.

Column	PBD				DBD		Additional Layer		Axon/AD	
	Input	Kernel size	Soma	Spikes$_o$	Array Number	Array Size	Spikes$_o$	Kernel Size	Array Number	Array Size
Two	32x3	2x2x4	15x1x4	15x1x4	15	4x12	None	None	5	60x10
Wider Single	43x3	2x2x4	21x1x4	21x1x4	21	4x12	None	None	7	84x10
Deeper Single	32x3	2x2x4	15x1x4	15x1x4	15	4x12	6x1x52	3x3x52	2	312x10

Table 2. Number of memristors used to form the DBD connections with different neighborhood ranges, and the corresponding accuracies

Neighborhood range	Memristors	Visual		Audio		Paired
		Training [%]	Test [%]	Training [%]	Test [%]	Test [%]
3	688	95.09	95.21	93.93	92.44	98.74
5	1104	95.10	95.00	93.95	93.56	98.94
7	1488	95.04	94.98	93.83	93.44	98.88
9	1840	95.02	95.03	94.19	93.22	98.71

Table 3. Classification accuracy for different grouping numbers, along with the number of memristors used for each case. Grouping significantly reduces the hardware size with minimal impact on accuracy.

Number of groups	Accuracy			Memristors
	Visual [%]	Audio [%]	Paired [%]	
1	95.96	92.44	98.88	900
3	95.21	92.44	98.74	300
5	92.85	91.78	98.21	180

Table 4. Average spike counts in Layer 1, for different network configurations.

Task	PBD w/o DWI	w/ PBD	w/ PBD+DBD	CLN
Visual	31.11	29.86	23.40	23.40
Audio	209.69	131.35	82.34	82.13

Table 5. The number of crossbar arrays used to implement different features for a single column in the proposed CLN.

Network	Number of Array	Array size
PBD	3	4x4
DBD	5	12x12
Axon (AD)	25	12x12

Table 6. Average energy consumption of different components in the CLN

Task	PBD			DBD	Axon/AD	Total
	E_{array}	E_{IF}	E_{DWI}	E_{array}	E_{array}	
	[pJ]	[pJ]	[pJ]	[pJ]	[pJ]	[pJ]
Visual	483	21.5	15.6	557	211	1290
Audio	2620	268	80.8	2280	1130	6390

Table 7. Performance metrics of the proposed CLN hardware system.

Number of PBD	Task	Total cycles	Latency [µs]	Power [µW]	IPS
1	Visual	616.41	12.33	104.51	81114
1	Audio	841.80	16.84	379.31	59367
5	Visual	206.14	4.12	312.52	242556
5	Audio	281.27	5.63	1135.24	177768
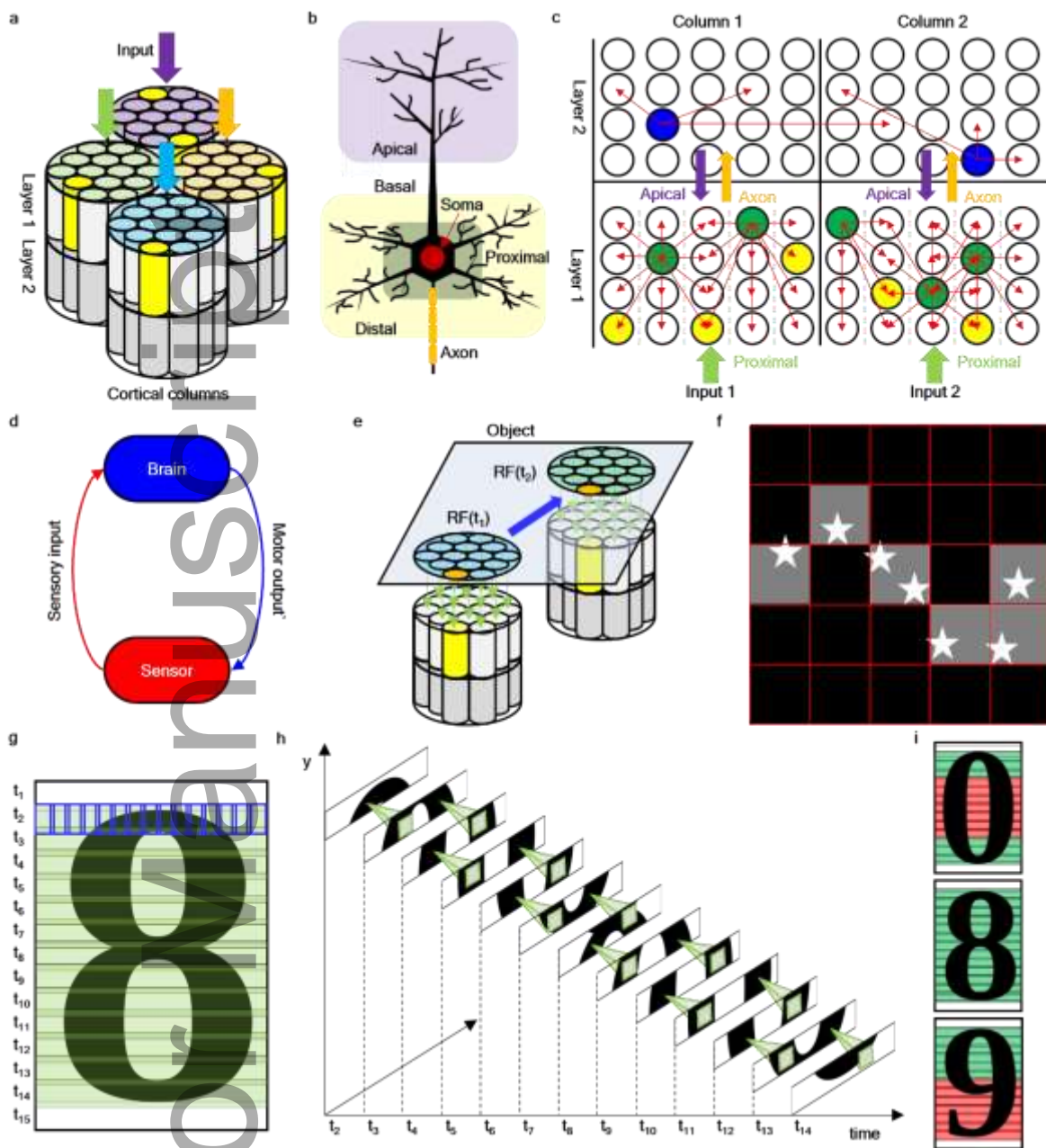
Figure 1. Columnar network structures. a) Columnar structures in the neocortex, showing 4 columns with 14 minicolumns each. b) Schematic of a pyramidal neuron. c) Different connections of the proposed CLN, showing PBD connections (green arrows) that feed the inputs to Layer 1 of the network, DBD connections (red arrows), axonal connections (orange arrows) and AD connections (purple arrows). d) Illustration the sensorimotor integration approach. e) The receptive field (RF) of a column. As the sensor moves, the RF is updated. A minicolumn (colored in yellow) may process a local region of the RF (marked by the orange circle). f) An example of the sensorimotor integration method to find stars in a constellation. g-i) Schematic of the CLN processing an input image. g) The RFs along time shown as green stripes. The local inputs processed by the minicolumns at t_2 are highlighted as blue rectangles as an example. h) the RFs sensed by the network at different timesteps. The firing neuron sends a prediction signal to neurons in its neighborhood at the next timestep (green squares). i) Firing patterns of the three candidate cells in Layer 2 fire over the 15 timesteps. Green: firing. Red: no firing.
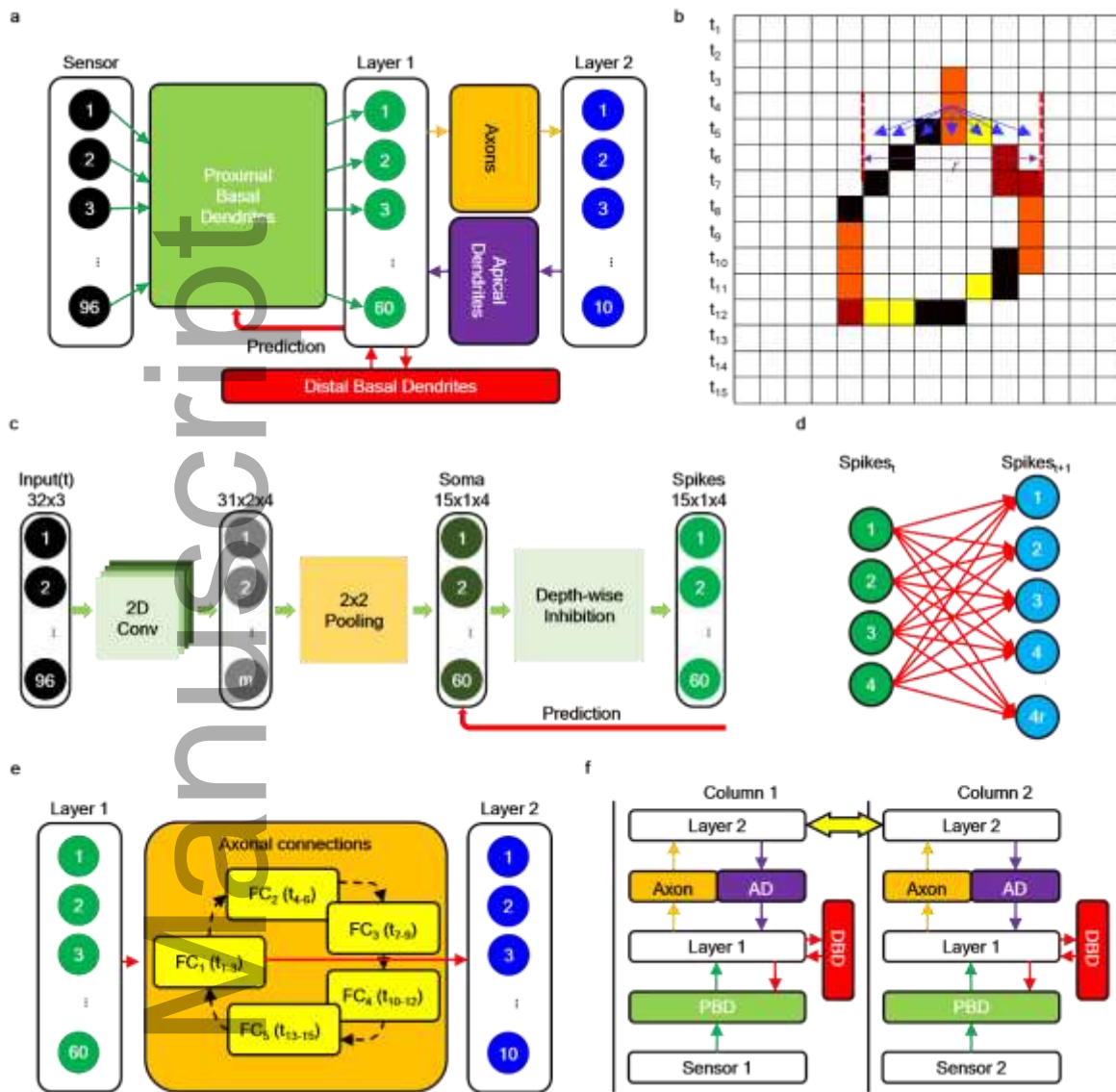
Figure 2. Proposed CLN configuration. a) Network configuration for a single column CLN, showing the PBD (green box), DBD (red box), axon (orange box) and AD (purple box) connections. b) Example input. The colors mark the membrane potential change after the PBD connections, and the blue arrows mark the DBD connections. c) PBD connection implementation. d) DBD connection from a single minicolumn, mapped as an FC layer. e) Axonal connections with several FC structures based on the grouping method. f) Schematic of a two-column CLN network, with each column receiving inputs from a different sensor.
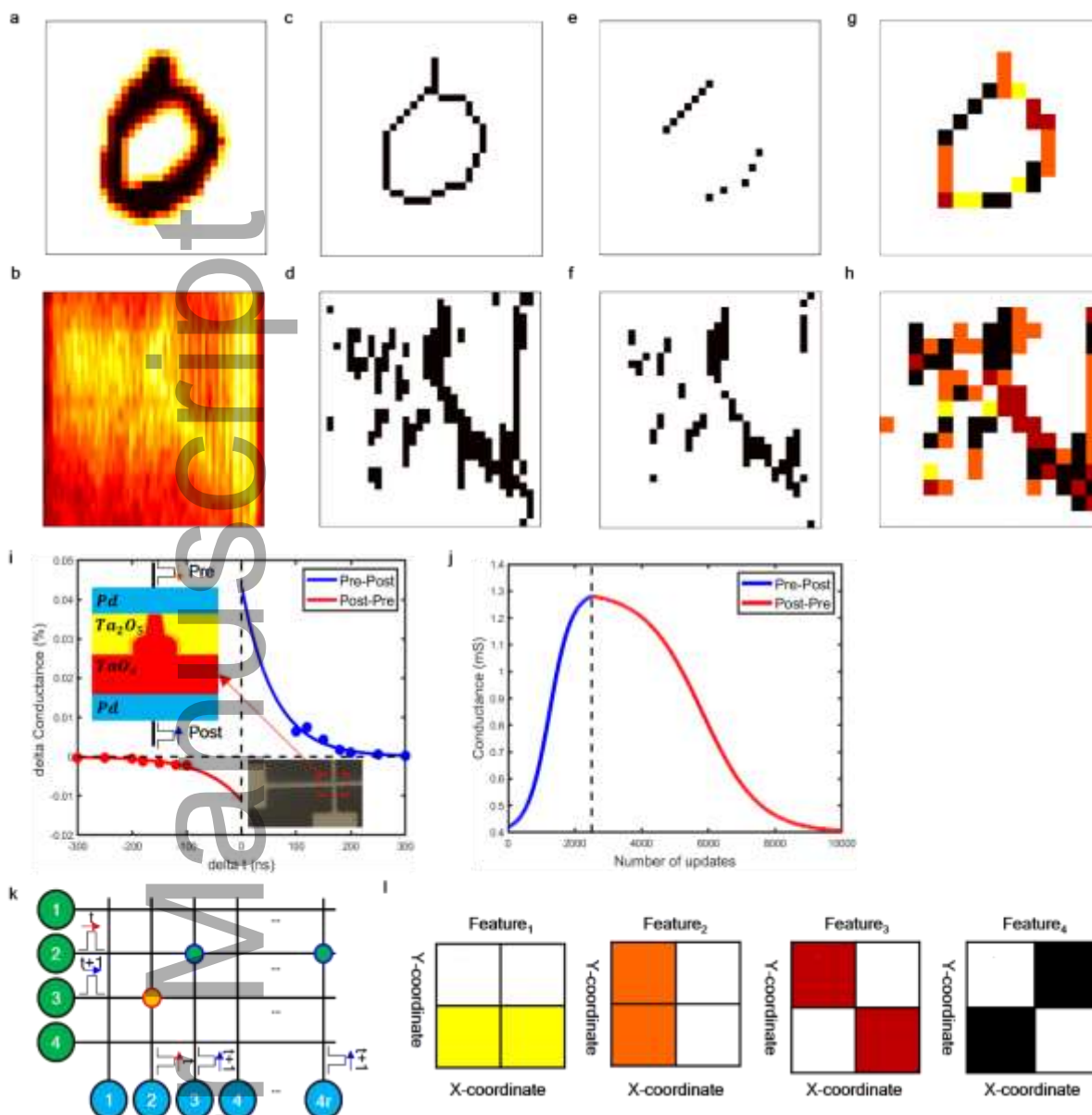
Figure 3. CLN for visual and audio input processing. a,b) An example of the original data of the same object in a) visual dataset and b) audio dataset. c,d) After preprocessing. e,f) After feedforward connections in the PBD layer. An example corresponding to output of the 4th feature in the convolution kernel is shown here. g,h) Firing neurons in the 15 minicolumns (plotted along the x-axis) after the PBD connections, during the 15 timesteps (plotted along the y-axis). The neurons are represented by their assigned colors in l). i) Measured data and fitting curves of a fabricated 2nd-order memristor natively implementing STDP. Lower nset: scanning electron microscope image of the fabricated device. Upper inset: schematic of the device structure. j) Simulated weight updates after the application of consecutive pulse pairs with fixed Δt of 150ns.[26] k) DBD connections implemented by a memristor crossbar. The DBD connections are learned by the local STDP rule natively implemented in the 2nd-order memristors, showing examples of potentiation (green dots) and depression (orange dot) l) The 4 2x2 convolution kernels used for the conv layer in the PBD connections. All colored squares are implemented by the low-resistance state of the memristor, and the white squares are implemented by the high-resistance state of the memristor.
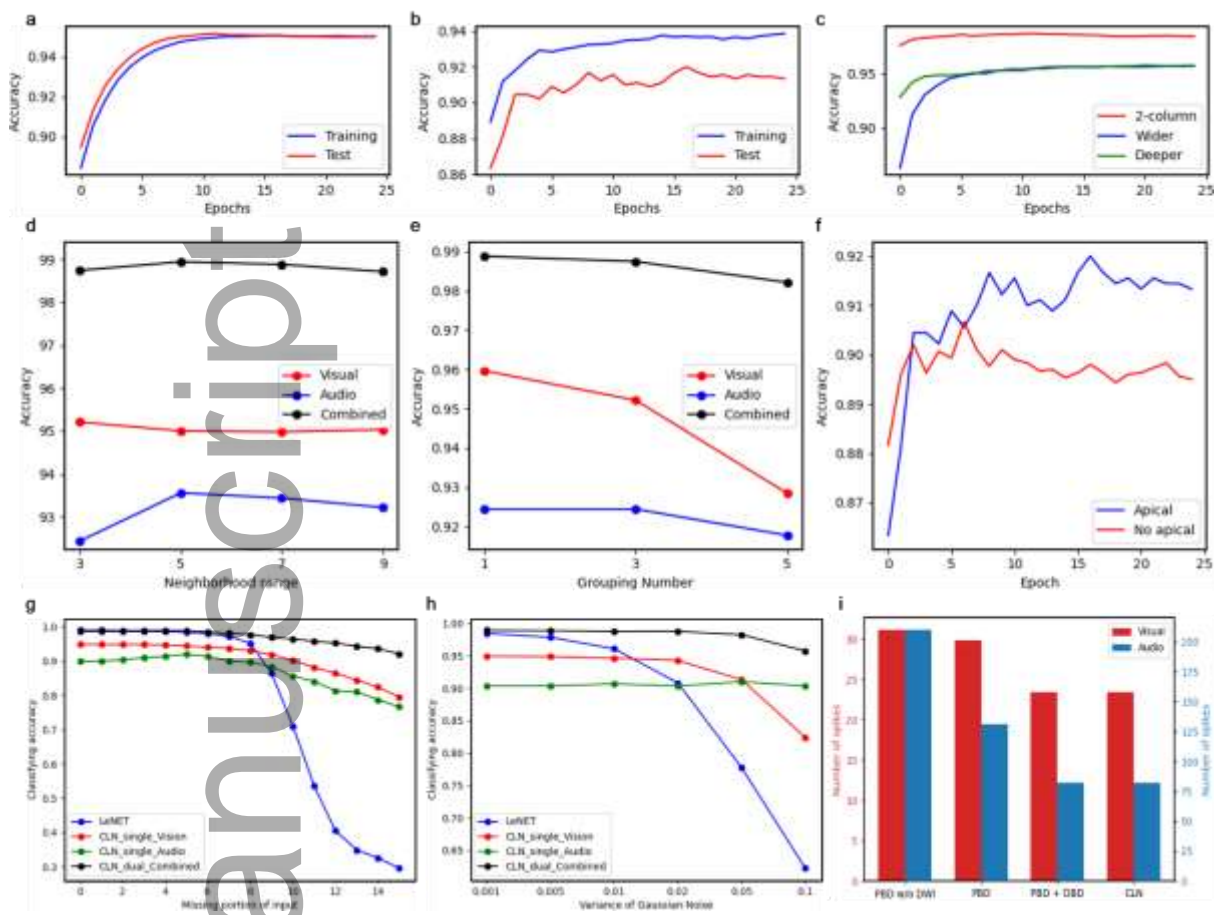
Figure 4. Performance of the proposed two-column CLN on visual, audio, and combined datasets. a,b) Classification accuracy for a) visual and b) audio datasets, during training and test. c) Test accuracy of two-column CLN for the combined dataset, along with test accuracy for the wider and deeper single-column CLNs using only visual inputs. d) Effects of the DBD connections' neighborhood range on test accuracy. e) Effects of grouping spikes from Layer 1 in the axonal connections on test accuracy. f) Effects of AD connections on accuracy. g,h) Effects of g) insufficient data and h) gaussian noise on the performance of different networks. i) Total number of Layer 1 neuron spikes averaged over the test dataset, for different network configurations.
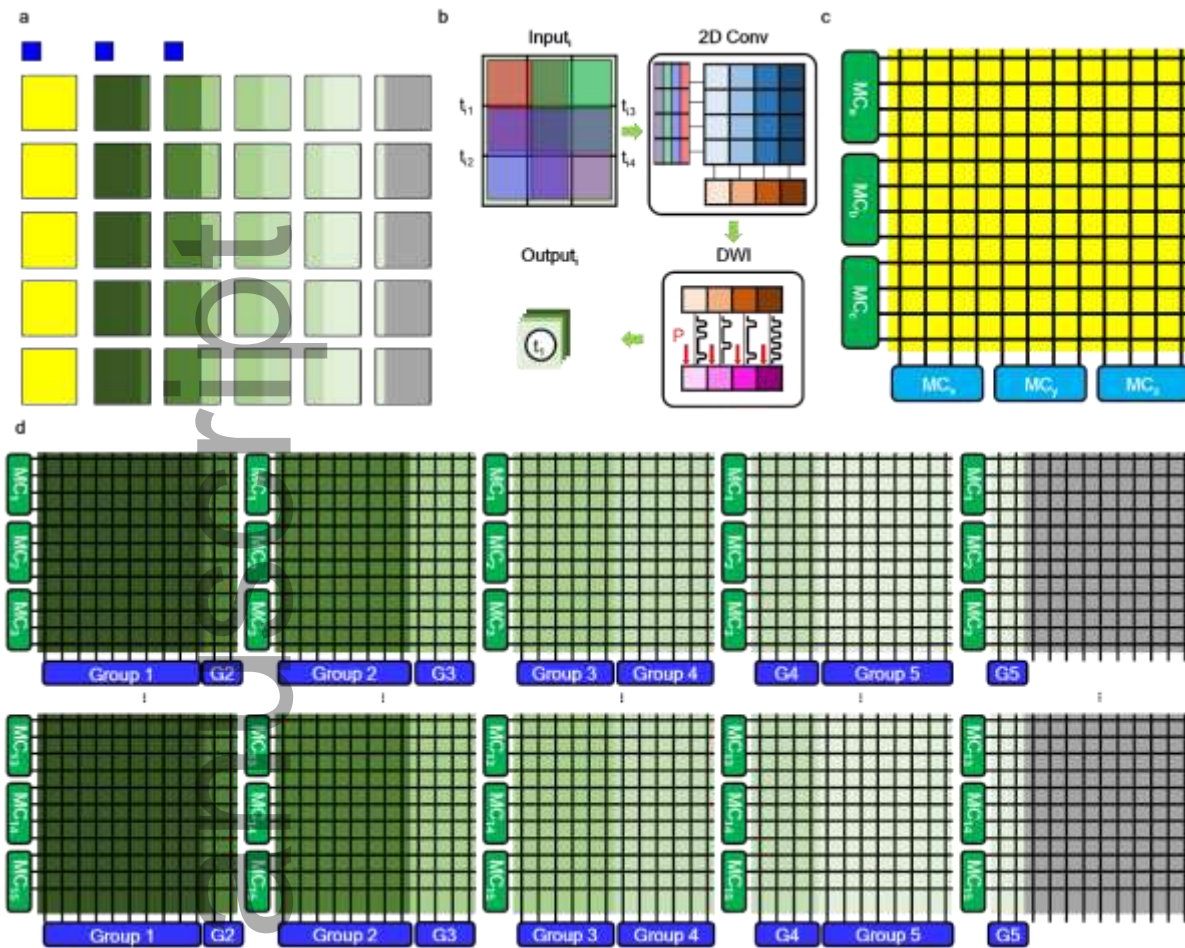
Figure 5. Hardware implementation. a) A column of the CLN implemented in a tiled memristor crossbar architecture. Blue blocks represent the convolution kernels of PBD connections, yellow and green blocks represent the DBD and axon/AD connections, respectively. b) Visualization of workflow of the PBD connections during four cycles (t_{i1}-t_{i4}). c) One of the 5 12x12 crossbars used to implement the DBD connections. Each minicolumn (MC_x, MC_y, MC_z) receives input spikes from 3 adjacent minicolumns. d) 25 12x12 crossbars used to implement the 5 60x10 axon/AD connections. The different shades of green represent the 5 groups during sensorimotor movement.

Reference

[1]   Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436.

[2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, *J. Big Data* **2021**, *8*, 53.

[3] S. Thorpe, A. Delorme, R. Van Rullen, *Neural Networks* **2001**, *14*, 715.

[4] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, T. Masquelier, *Pattern Recognit.* **2019**, *94*, 87.

[5] P. Diehl, M. Cook, *Front. Comput. Neurosci.* **2015**, *9*, 99.

[6] S. Ghosh-dastidar, A. G. Lichtenstein, *Int. J.* **2009**, *19*, 295.

[7] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, *Neural Networks* **2019**, *111*, 47.

[8] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, J. Alakuijala, in *ICASSP 2020 - 2020 IEEE Int. Conf. Acoust. Speech Signal Process.*, **2020**, pp. 8529–8533.

[9] R. Brette, *Front. Syst. Neurosci.* **2015**, *9*.

[10] M. A. Zidan, Y. J. Jeong, J. Lee, B. Chen, S. Huang, M. J. Kushner, W. D. Lu, *Nat. Electron.* **2018**, *1*, 411.

[11] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, W. D. Lu, *Nat. Electron.* **2019**, *2*, 290.

[12] S. Chen, M. R. Mahmoodi, Y. Shi, C. Mahata, B. Yuan, X. Liang, C. Wen, F. Hui, D. Akinwande, D. B. Strukov, M. Lanza, *Nat. Electron.* **2020**, *3*, 638.

[13] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, H. Qian, *Nature* **2020**, *577*, 641.

[14] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, *521*, 61.

[15] S. Yin, J. S. Seo, Y. Kim, X. Han, H. Barnaby, S. Yu, Y. Luo, W. He, X. Sun, J. J. Kim, *IEEE Micro* **2019**, *39*, 54.

[16] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C. Xue, W. Chen, J. Tang, Y. Wang, M. Chang, H. Qian, H. Wu, *2020 IEEE Int. Solid- State Circuits Conf. -* **2020**, 500.

[17] G. C. Adam, B. D. Hoskins, M. Prezioso, F. Merrikh-Bayat, B. Chakrabarti, D. B. Strukov, *IEEE Trans. Electron Devices* **2017**, *64*, 312.

[18] S. Yin, X. Sun, S. Yu, J.-S. Seo, *IEEE Trans. Electron Devices* **2020**, *67*, 4185.

[19] V. B. Mountcastle, *Brain* **1997**, *120*, 701.

[20] N. Spruston, *Nat. Rev. Neurosci.* **2008**, *9*, 206.

[21] J. Hawkins, S. Ahmad, Y. Cui, *Front. Neural Circuits* **2017**, *11*.

[22] J. Hawkins, S. Ahmad, *Front. Neural Circuits* **2016**, *10*, 1.

[23] X. Li, J. Tang, Q. Zhang, B. Gao, J. J. Yang, S. Song, W. Wu, W. Zhang, P. Yao, N. Deng, L. Deng, Y. Xie, H. Qian, H. Wu, *Nat. Nanotechnol.* **2020**, *15*, 776.

[24] Y. J. Jeong, S. Kim, W. D. Lu, *Appl. Phys. Lett.* **2015**, *107*, DOI 10.1063/1.4934818.

[25] S. Kim, C. Du, P. Sheridan, W. Ma, S. Choi, W. D. Lu, *Nano Lett.* **2015**, *15*, 2203.

[26] S. Yoo, Y. Wu, Y. Park, W. D. Lu, *Adv. Electron. Mater.* **2022**, *Issued onl*, 2101025.

[27] E. G. Jones, *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 5019.

[28] G. Rinkus, *Front. Neuroanat.* **2010**, *4*.

[29] P. Rakic, *Nat. Rev. Neurosci.* **2009**, *10*, 724.

[30] D. P. Buxhoeveden, M. F. Casanova, *Brain* **2002**, *125*, 935.

[31] M. Megías, Z. Emri, T. F. Freund, A. I. Gulyás, *Neuroscience* **2001**, *102*, 527.

[32] W.-C. A. Lee, V. Bonin, M. Reed, B. J. Graham, G. Hood, K. Glattfelder, R. C. Reid, *Nature* **2016**, *532*, 370.

[33] H. Ko, S. B. Hofer, B. Pichler, K. A. Buchanan, P. J. Sjöström, T. D. Mrsic-Flogel, *Nature* **2011**, *473*, 87.

[34] Y. Yoshimura, H. Sato, K. Imamura, Y. Watanabe, *J. Neurosci.* **2000**, *20*, 1931.

[35] J. Park, A. Papoutsi, R. T. Ash, M. A. Marin, P. Poirazi, S. M. Smirnakis, *Nat. Commun.* **2019**, *10*, 5372.

[36] J. J. Nassi, S. G. Lomber, R. T. Born, *J. Neurosci.* **2013**, *33*, 8504.

[37] C. Wang, W. J. Waleszczyk, W. Burke, B. Dreher, *Exp. Brain Res.* **2007**, *182*, 479.

[38] P. Anderson, H. Silfvenius, S. H. Sundberg, O. Sveen, *J. Physiol.* **1980**, *307*, 273.

[39] N. L. Golding, T. J. Mickus, Y. Katz, W. L. Kath, N. Spruston, *J. Physiol.* **2005**, *568*, 69.

[40] J. K. O'Regan, A. noë, *Synthese* **2001**, *129*, 79.

[41] M. Flanders, *Biol. Cybern.* **2011**, *104*, 1.

[42] R. Shadmehr, M. A. Smith, J. W. Krakauer, *Annu. Rev. Neurosci.* **2010**, *33*, 89.

[43] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, W. D. Lu, *Nat. Commun.* **2017**, *8*, 1.

[44] J. Moon, W. Ma, J. H. Shin, F. Cai, C. Du, S. H. Lee, W. D. Lu, *Nat. Electron.* **2019**, *2*, 480.

[45] L. Deng, *IEEE Signal Process. Mag.* **2012**, *29*, 141.

[46] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, A. Thite, "Jakobovski/free-spoken-digit-dataset: v1. 0.8.," can be found under http://github.com/Jakobovski/free-spoken-digit-dataset, **2018**.

[47] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, T. Masquelier, *Front. Neurosci.* **2019**, *13*, 1.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, in *Adv. Neural Inf. Process. Syst.* (Eds.: H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, R. Garnett), Curran Associates, Inc., **2019**.

[49] N. Spruston, D. B. Jaffe, D. Johnston, *Trends Neurosci.* **1994**, *17*, 161.

[50] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, *Proc. IEEE* **1998**, *86*, 2278.

[51] X. Wang, Q. Wang, F.-H. Meng, S. H. Lee, W. D. Lu, in *2020 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst.*, **2020**, pp. 141–144.

[52] X. Wang, R. Pinkham, M. A. Zidan, F.-H. Meng, M. P. Flynn, Z. Zhang, W. D. Lu, *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 559.

[53] J. A. Starzyk, Y. W. Jan, *Midwest Symp. Circuits Syst.* **1996**, *1*, 501.

[54] F.-X. Liang, I.-T. Wang, T.-H. Hou, *Adv. Intell. Syst.* **2021**, *3*, 2100007.

[55] S. Yin, S. K. Venkataramanaiah, G. K. Chen, R. Krishnamurthy, Y. Cao, C. Chakrabarti, J. S. Seo, *2017 IEEE Biomed. Circuits Syst. Conf. BioCAS 2017 - Proc.* **2018**, *2018-Janua*, 1.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, *J. Mach. Learn. Res.* **2011**, *12*, 2825.

[57] P. Falez, P. Tirilly, I. M. Bilasco, *Proc. Int. Jt. Conf. Neural Networks* **2020**, 0.

TOC Graphic:
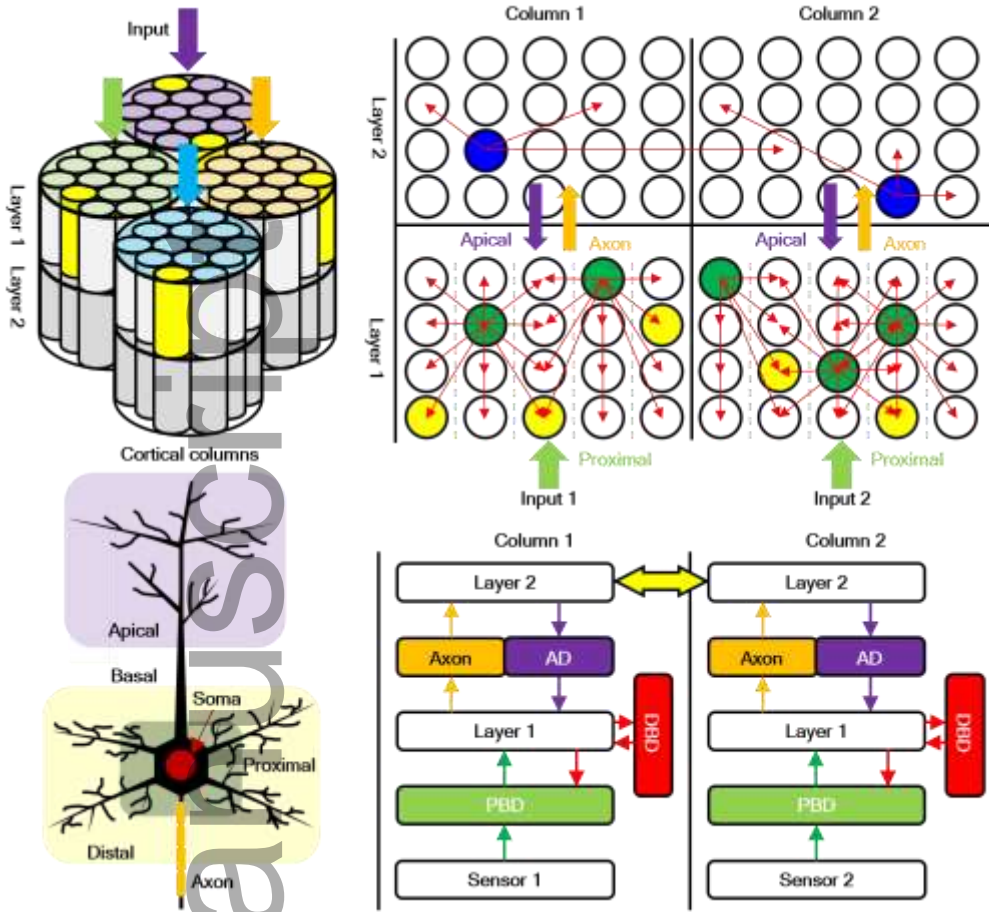
Inspired by column structures in the mammalian cortex, a Columnar Learning Network was proposed and demonstrated using 2nd-order memristors. The CLN can perform robust classification with sparse data, while learning spatiotemporal correlations in inputs from different regions of an object. Multi-sensor processing enabled by CLN leads to higher classification accuracy and better noise tolerance.