Hierarchical Temporal Memory Features with Memristor Logic Circuits for Pattern Recognition

Olga Krestinskaya, *Graduate Student Member, IEEE*, Timur Ibrayev, *Student Member, IEEE*, and Alex Pappachen James, *Senior Member, IEEE*

***Abstract*—Hierarchical temporal memory (HTM) is a machine learning algorithm inspired by the information processing mechanisms of the human neocortex and consists of a spatial pooler (SP) and temporal memory (TM). In this paper, we develop circuits and systems to achieve the optimized design of an HTM SP, an HTM TM, and a memristive analog pattern matcher for pattern recognition applications. The HTM SP realizes an optimized hardware design through the introduction of mean overlap calculations and by replacing the threshold determination in the inhibition stage with a weighted summation operator over the neighborhood of the pixel under consideration. HTM TM is based on discrete analog memristive memory arrays and a weight update procedure. The operation of the proposed system is demonstrated for a face recognition problem, using the standard AR, ORL, and Yale databases, and for speech recognition, using the TIMIT database, with achieved accuracies of 87.21% and approximately 90%, respectively, given an SNR of 10 dB. Visual data processing using binary HTM SP features requires less storage and processing memory than required by the traditional processing methods, with the area and power requirements for its implementation being 0.096 mm^2 and 1756 mW, respectively. The design of the TM circuit for a single pixel requires 23.85 μm^2 of area and 442.26 μW of power.**

***Index Terms*—CMOS, face recognition, hierarchical temporal memory (HTM), HTM features, memristors, spatial pooler (SP), template matching, temporal memory (TM).**

I. Introduction

HIERARCHICAL temporal memory (HTM) is a machine learning algorithm that attempts to mimic human neocortex operations [1]. The HTM architecture consists of a spatial pooler (SP) and temporal memory (TM) and is characterized by sparsity, modularity, and hierarchy [2]. The HTM SP performs sparse distributed representation of the input data, while the HTM TM is responsible for the learning process [3]. The imitation of human brain functionality makes HTM a core-adjustable algorithm applicable to various operations such as visual data processing [4], object categorization [5], pattern discovery [6], and data classification [7].

Although HTM-based algorithms have applications in image processing [3], [8], [9], the hardware circuits for HTM SP and HTM TM implementation remain an open-ended research question [10]. Hardware-level implementation is essential to allow sensor-level integration of HTM circuits and systems that has the potential to be included in modern Internet of Things applications. The smaller area and lower power requirements make such sensor integration an important topic to address regarding the intelligent processing of the ever-increasing requirements on the volume, veracity, and versatility of sensory data. We propose a set of circuits for building a face and speech recognition system consisting of three main parts: 1) the HTM SP; 2) the HTM TM; and 3) a memristive pattern matcher. The HTM SP is used for the extraction of important face and speech features. The HTM TM participates in the learning process during system training. The memristive pattern matcher is applied for feature comparison and final decision making during the recognition stage.

In contrast to our previous work on HTM [11], the following work proposes a modification of the HTM SP that includes the replacement of the conventional [winner-takes-all (WTA)-based] threshold calculation in the inhibition phase of the HTM SP to the threshold calculation with a weighted summation operator considering the neighborhood of the pixel. The proposed HTM TM concept is then applied for the learning operation during the training phase of the recognition operation and is based on the weight update procedure. The hardware implementation of the HTM TM includes the discrete analog memristive memory array consisting of the memristive memory cells and the circuit for the weight update process.

Although the work presented in [11] proposed memristive crossbar circuits for the conventional HTM SP and enabled the compact storage and parallel processing of synapses, it established certain limitations to processing speed. Ensuring that the design remained similar to conventional HTM SP algorithms required circuits presented in [11] to process input space block by block in a serial manner, thereby, increasing the total processing time. Hence, the new circuits presented in this paper and the resulting novelties, expressed in terms of modifications of the HTM SP as well as the addition of

Manuscript received March 15, 2017; revised June 26, 2017; accepted August 13, 2017. Date of publication August 31, 2017; date of current version May 18, 2018. This paper was recommended by Associate Editor Y. Shi. *(Corresponding author: Alex Pappachen James.)*

O. Krestinskaya is with the Bioinspired Microelectronics Systems Group, Nazarbayev University, Astana 01000, Kazakhstan.

T. Ibrayev was with the Bioinspired Microelectronics Systems Group, Nazarbayev University, Astana 01000, Kazakhstan. He is now with Purdue University, West Lafayette, IN 47907 USA.

A. P. James is with the Bioinspired Microelectronics Systems Laboratory, School of Engineering, Nazarbayev University, Astana 01000, Kazakhstan (e-mail: apj@ieee.org).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The file contains the formal description of HTM spatial pooler and temporal memory algorithms. The total size of the file is 190 kB.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2748024

analog HTM TM, were required to be introduced to optimize the overall processing of the HTM system. As a result, a better performance of the HTM SP as well as easier hardware implementation and circuit realization were achieved.

The constructed hardware architecture for face and speech recognition has an advantage in terms of the required memory space for data processing and improved processing speed. The traditional hardware for visual data processing is designed for processing analog signals, which requires a large amount of memory. If the conversion from analog signals to digital is required, analog-to-digital converters are applied [12]. When the converted signal is input to co-processors or FPGAs, the processing speed and sampling rate are significantly decreased. In contrast, HTM overcomes this problem and goes beyond traditional computing methods. The proposed HTM SP is able to convert analog signals into digital signals without using analog-to-digital converters. The signal after HTM SP processing is completely digital; this enables a required memory space for data processing. In addition, because the HTM SP avoids using analog-to-digital converters, the processing speed and sampling rate are increased. The HTM TM part is also purely analog, which allows the analog data to be stored in a memristive memory array without converting the data into digital form and training the system without application of additional algorithms and computing software.

The main contributions of this paper include the following.

1) We report a new design of a scalable face recognition system based on the HTM principle and implemented with CMOS-memristive analog circuits for generating and comparing image features in a parallel, hierarchical and modular manner.
2) We present practical solutions to reduce the number of hardware components required for face and speech matching by creating HTM SP face and speech templates.
3) We develop a simplified approach to implement the HTM SP circuit based on a memristive averaging circuit.
4) We introduce the HTM TM concept and its purely hardware implementation based on the discrete analog memristive memory array and memristive-CMOS circuit for updating the stored data.
5) We conduct extensive experimental studies and circuit simulations on benchmark data sets, which verify the usefulness of our proposed approach.

This paper is organized into six sections. Section II describes conventional HTM and related works. Section III proposes a modified HTM SP and HTM TM model. Section III-B introduces the overall architecture of the face recognition system and illustrates the circuit implementation of the system components. In Section V, a discussion of the proposed system and the obtained results are given. Section VI concludes this paper.

II. Conventional HTM

HTM was inspired by the neocortex functions and describes the overall theory outlined in the book *On Intelligence* [13]. The HTM theory was developed based on several functional and structural biologically relevant observations of the neocortex. These observations included the structure and functional hierarchy of information processing within the neocortex, the generality of the neocortex algorithms, the ability to process features in a sparse, distributed manner, the layered process of extracting and deciphering complex information, the real-time encoding of sensory information, the dynamic streaming and sequencing of memories for data processing, and the ability to process data online. These principles are incorporated into HTM algorithm design, which consists of two distinct modules: 1) SP [14] and 2) TM [15].

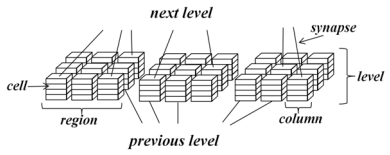

Fig. 1. Single level within the hierarchy of HTM.

The design and architecture of HTM are based on the modeling of pyramidal neurons in the cortex and are very different from those of the neuron models used in artificial neural networks. The model permits active dendrites for the recognition of independent patterns from a large population of cells. The ability to integrate synapses into the model enables the prediction of the sequence transition of cell activities. Furthermore, the HTM architecture relies on the principles of Hebbian learning, the network connectivity of sensory cortices, homeostatic excitability control, and the structural plasticity, which depends on the activity.

The HTM algorithm is organized into regions comprising columns of cells, as shown in Fig. 1, with each column of cells forming a single computational unit in which SP operates. The mechanism of emulating the sparse activation of neurons is referred to as inhibition, which is the ability of columns in SP to inhibit neighboring columns within an inhibition radius (size of the local neighborhood) from becoming active. The junctions between cells are called synapses, and the synapses on a columns dendritic segment enable connecting to the bits in the input space. The receptive field is the available input space in which the columns can connect, and the permanence value is a measure of growth between columns and one of the cells in the receptive field. If the value of a synapse's permanence is above a permanence threshold, it is considered to be fully connected.

Since HTM is a cortical algorithm, it provides a new model and direction for the hardware implementation of neocortical functions [16]. The robustness and feasibility of HTM have been tested using image processing, object categorization, and recognition applications [5]. These algorithmic implementations are promising. Melis *et al.* [7] and Deshpande [17] depicted early digital designs for the VLSI architecture and FPGA implementation of HTM, respectively. The analog design implementations include a memristor-based design [18] and scalable memristor crossbar architectures [11] for the SP and a mixed-signal design of spin neurons and a crossbar-based implementation [19] for both the SP and TM.

The SP on its own is capable of learning and classifying different data sets, such as numbers, letters, and pixels [20]. Thus, the implementation of the SP alone in applications such as image processing, pattern recognition, and speech recognition yields good performances. The SP is realized via

implementation in four phases: initialization, overlap computation, inhibition, and learning [2]. The steps for implementing the SP are as follows.

1) The SP accepts input data bits from sensory data or from other regions of the HTM.
2) The HTM regions are initialized by selecting a fixed number of columns that can receive inputs. Each column is connected to the input using a dendrite segment having a set of potential synapses. The potential synapse and its corresponding permanence value are initialized randomly around the permanence threshold. Some of these potential synapses with a permanence value greater than the permanence threshold will already be connected.
3) The number of synapses on each column connected to active (ON) input bits is calculated; these connected synapses are referred to as active synapses.
4) These active synapses are multiplied by a boostingİ factor. The boosting factor represents the frequency of activeness of the column relative to its neighborhood.
5) Within the inhibition radius, the columns with the highest activations become active, while the others are disabled. Since the inhibition radius depends on the spread of input bits, the column activations are sparsely distributed.
6) The SP region follows a Hebbian-style learning rule to update the permanence values of all the potential synapses; thus, the synapses are changed from being connected to being unconnected and vice-versa.
7) Step 3 is repeated for subsequent inputs.

The activation rule in step 5 is implemented using the k-winners-take-all computation within a local neighborhood. Ideally, the parameter k can be adjusted to regulate the desired number of winning columns [2]. However, in [11] and [18], the inhibition phase is implemented by the WTA circuit; as a result, the value of the desired activity level is limited to 1. A more formal mathematical description of SP is provided in the supplementary material.

The second part of HTM is the TM. Whereas the SP is responsible for the sparse representation of the input data, the TM considers the changes that occur over time, learns the patterns and attempts to make predictions based on the history of input information. The TM aims to learn the connections between cells within the same layer, while the SP aims to learn the feed-forward connections between input bits and columns. The TM algorithm is also known as the sequence memory. The main roles of the TM are the following: 1) to learn sequences of active columns from the SP over time and 2) to predict which patterns come next based on the temporal context of each input.

The TM collects the input from the SP, with the feed-forward inputs of the TM originating from the active columns. The steps for implementing the TM are as follows.

1) Activate the cells in each of the active columns that are in the predictive state. If none of the active columns are in the predictive state, then activate all of the cells in the column. These active cells now represent input related to prior input.
2) Find the total number of synapses connected to active cells for all the dendrite segments of every cell in a given layer. If this number exceeds a threshold, the corresponding dendrite segment is assigned to be active. The cells corresponding to this dendrite segment are set to the predictive state unless already set because of the feed-forward input. The collection of cells in the predictive state represents the predicted pattern for the layer.
3) The activation of a dendrite segment is required to update the permanence values of the synapses in the given segment. The permanence values of the potential synapse are increased for active cells and decreased for inactive cells. These modifications are temporarily marked, and the synapses on already trained segments are made active, leading to prediction.
4) The feed-forward inputs can change the cell state from inactive to active (if this occurs, the temporary marks are removed) and affect each potential synapse of the cell. In other words, the permanence of synapses is only updated in the case of the correct prediction of feed-forward activation of a cell.
5) On the other hand, to change the cell state from the predictive state to the inactive state, we need to undo any permanence changes marked as temporary for each potential synapse. When one cell incorrectly predicts the feed-forward activation of another cell, the permanence values of the previously active synapse are decreased.

The predictive state in the TM is purely an internal state of the cell. The active cells resulting from the feed-forward input propagate their activity to avoid the chain of predictions leading to further predictions. A more formal mathematical description of the TM is provided in the supplementary material.

III. Proposed HTM Architecture

In this paper, we differentiate the conventional HTM algorithm that is discussed in the previous section from the algorithm that we propose for implementation. The latter algorithm will be referred to as the modified HTM algorithm. We highlight the major aspects of the proposed modifications in following Section III-A. The circuit designs as well as the system-level algorithms for modified HTM are presented in Section III-B and supplementary material provides additional algorithmic representation of the operating principle of the proposed system.

A. Overview of Proposed Architecture

1) Spatial Pooler: The primary difference from the aforementioned HTM algorithm is that we are changing the criteria for the selection of winning columns that occurs in the inhibition phase. Instead of considering the k-th largest overlap value, we propose to calculate the threshold and establish the selection of the winning columns based on the average value of the *overlaps* in the modified SP.

The implementation of the proposed SP design is discussed in Section III-B and requires consideration of the availability of resources. Two types of resources being regarded in the hardware design of interest are processing speed and on-chip area. The user preferences concerning the above-mentioned resources dictate whether the operations will be

performed in parallel or serially. The former implementation requires concurrent replication of the circuit for each of the inhibition blocks such that the processing of an image is performed at once. This benefits the design in terms of fast processing. However, a high on-chip area demand is suffered. Alternatively, serial implementation would require a single module for the processing of one inhibition block after another, which would reduce the required on-chip area. Thus, tradeoffs between high-speed processing and a low-area network should be made.

2) Temporal Memory: In this paper on TM, we use a single training image known as a *class map*. The class map considers the training images within the same class and combines all of their features. This allows the matching of the patterns in each of the trained classes to be executed through the comparison of the testing image with a single image. As a result, the memory requirements as well as the time for the processing are reduced.

Such a design of TM is based on two properties: 1) focusing and 2) reflection. The fact that the circuit of TM is established after the realization of the SP shifts the *focus* of the learning procedure onto the important and unimportant features of the original training images. The *reflection* property is related to the changes in the features over time. This is realized by considering the importance of the input bits and introducing corresponding changes to the weights of the TM cells.

The weights of the TM cells can be incremented or decremented by the weight update value $\pm\Delta$. The positive weight update procedure, i.e., increasing the weight by $+\Delta$, occurs when the input bit represents an important bit, in other words, when the input bit from the feature extracted image is 1. A 0 value of the input bit of the feature extracted image represents an unimportant feature and will contribute to the negative update of the weight $(-\Delta)$. Thus, in contrast to the process of the formation of the feature extracted images, which used binary weights with values of either 1 or 0, the learning process utilizes the multivalued weights of the TM cells.

The formation of the class map for one of the image classes is demonstrated on Fig. 2. The inputs to the TM are the multiple feature extracted binary images, while the output is a single analog image of the same size that combines all the important and unimportant features. The training sequence that is applied in the TM eliminates the memory cells that initially had the same weights.

B. Circuit Design of Modified HTM Architecture

1) System Overview: The overall design of the proposed face recognition system is shown in Fig. 3. The system consists of an input data controller with data storage, the HTM SP, an output data controller, HTM TM and a memristive pattern matcher. The images from the capturing device, such as a camera, are sent to the input data controller, which selects the required number of images and stores them in the input data storage. Depending on the capacity of the system, the size of the input images and the number of available on-chip resources, the input data controller selects a parallel or serial processing type and rearranges the images into blocks, followed by preprocessing and filtering. Then, the testing image or parts of the image are processed by the proposed modified

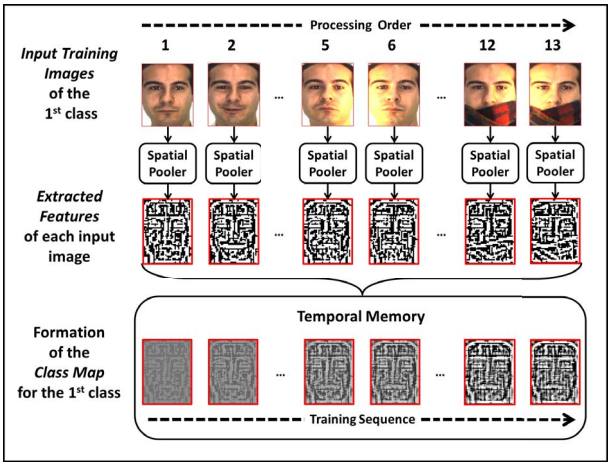

Fig. 2. Underlying principle of single-class-map formation using TM and feature extracted images obtained from the SP.

HTM SP, which performs feature extraction and produces the binary output image containing only the most relevant facial features. Next, the HTM SP output image is sent to the output data controller, which routes the images to HTM TM or the memristive pattern matcher according to the training or testing mode, respectively. Note that the test images in the testing mode and the training images in the training mode are separate sets of images.

During the training mode, the output data controller saves the image with the corresponding class number into the TM. When a new training image of the same class arrives, the training template of this class, called the *class map*, is updated according to the proposed HTM TM algorithm. In the testing mode, the output data controller directs the image into the memristive pattern matcher and retrieves the class maps of each class from TM. The memristive pattern matcher compares the testing image with all class maps and determines the similarity score for each comparison based on XOR logic and averaging operations. The class of the image corresponds to the match with the minimum difference between the testing image and the training class map (or the maximum similarity score) and is determined using a WTA circuit.

2) Spatial Pooler: Fig. 4 illustrates the circuit diagram of a single processing block of the proposed SP of the modified HTM. The process of extracting features from an input image is achieved by initially reading input bits (in yellow) from the input space (in purple) by processing the entire image in a block-by-block manner. Each of such blocks on the input space, which are called inhibition regions (in red), circumscribes the input bits constituting different RB regions (blue boxes in the top-right inhibition region). A single processing block, illustrated in Fig. 4, processes a single inhibition region and determines which bits within it are important and which are not important.

Fig. 5 illustrates the structure of a single RB, which performs an operation similar to that of a single HTM column in HTM theory (hence, both of the terms will be used interchangeably). An illustrated exemplar RB has in total $N \times K$ *random weight synapses*, each of which is implemented by a memristor device, and the permanence value of the block

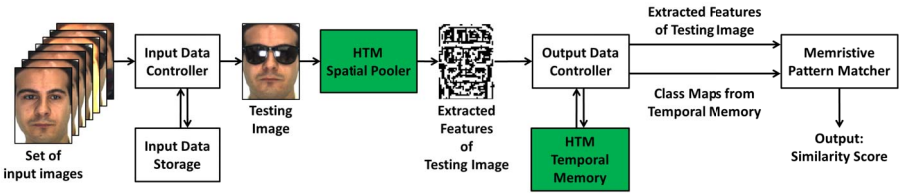

Fig. 3. High-level block diagram of the proposed pattern recognizer based on Modified HTM. The pattern recognizer consists of an input data controller for captured image storage and preprocessing, an HTM SP for feature extraction, HTM TM for training of the recognizer, an output data controller to control switching between train and testing modes and a memristive patter matcher used for image classification.

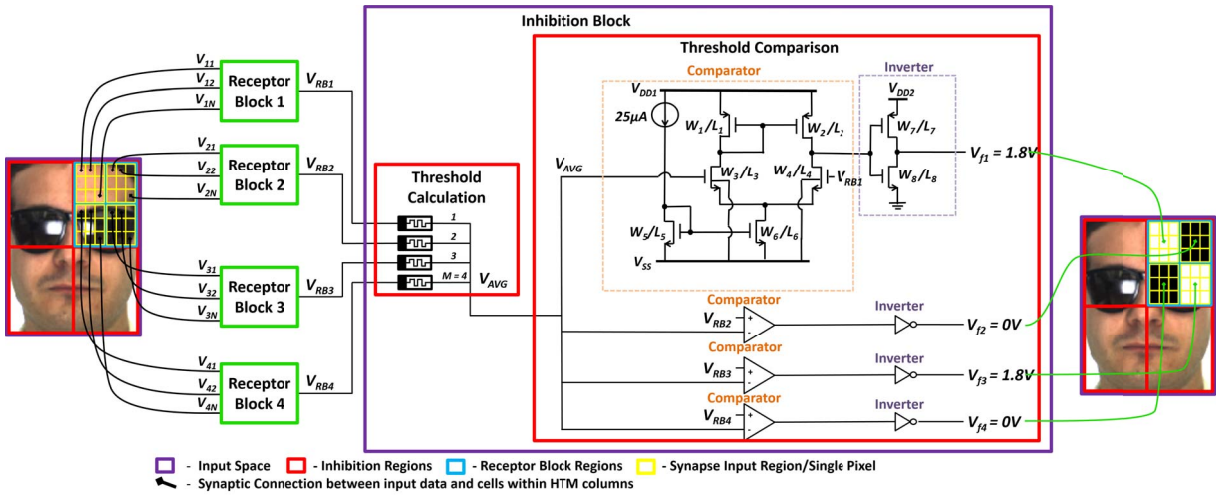

Fig. 4. HTM SP configuration. The processed image is divided into receptor blocks (RBs) consisting of N image pixels. M RBs form a single inhibition block. An inhibition block consists of two main parts: a thresholding calculation block and a threshold comparison block. The inhibition block produces a binary output.

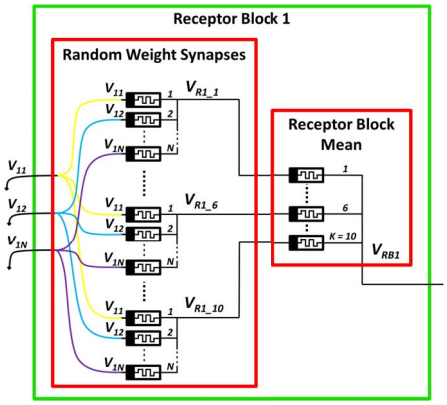

Fig. 5. Structure of a single RB, illustrating parallel synaptic connections represented by $N \times K$ memristors and a block calculating mean of those input synapses, which represents the output of the single RB.

corresponds to the memristance value of the device. A synaptic connection is counted as either connected or disconnected based on whether the memristance value is high (R_H) or low (R_L), respectively. If there are M RBs within a single processing block, as illustrated in Fig. 4, then within each of these RBs, the group of N synaptic connections having the same index number k such that $k \in [1, K]$ acquire the same set of random weights. According to the proposed design, the parameter K is chosen to have a fixed value $K = 10$, representing ten repetitions of the same input data.

Hence, the repetition operation is achieved when the bits of a single RB region are fetched by a corresponding RB. Specifically, the voltage signals representing input bits are applied to the memristors of a particular RB j such that $j \in [1, M]$ to produce the output signal $V_{\mathrm{RB}j}$, which represents the averaged weighted sum of its input bits. As illustrated in Fig. 5, the average within each RB (designated as the RB *mean*) is determined by memristor devices having the same high memristance value, R_H. According to HTM theory, the output signal $V_{\mathrm{RB}j}$ can then be said to represent the overlap value of the column j, which indicates the importance of the bits connected to it in comparison with the bits connected to other columns within the inhibition region also having M columns.

Next, as illustrated by the *threshold calculation block* in Fig. 4, based on the output voltage value of each RB, the average value V_{AVG} is calculated for the entire inhibition region. As illustrated by the *threshold comparison block* in Fig. 4, this average value V_{AVG} is used as a reference in determining which bits are important and which are not important within a single inhibition block. If the output overlap value $V_{\mathrm{RB}j}$ of an RB j is higher than the average value V_{AVG}, then the bits connected to that RB are counted as important. This, in turn, results in the output voltage signal having the value of $V_{fj} = 1.8$ V. Selecting a 180 nm TSMC CMOS technology, the voltage 1.8 V corresponds to logic "1," while $V = 0$ V refers to logic "0." In contrast, when the overlap value $V_{\mathrm{RB}j}$ is less than the average value, the bits connected to that RB

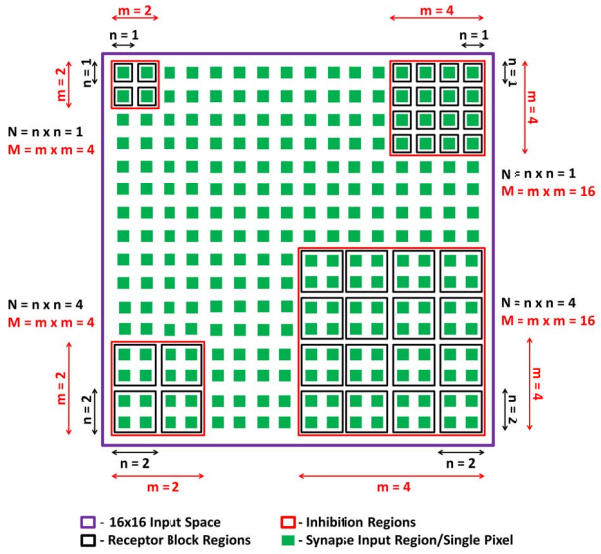

Fig. 6. Exemplary image consisting of $16 \times 16 = 256$ bits along with four different possibilities of how the image can be processed.

are counted as unimportant, and the output voltage signal will be $V_{fj} = 0$ V.

By processing all the inhibition regions of the input image in the same way, a binary feature extracted image is produced. A feature extracted image will have white bits representing important features and black bits representing the remainder of an image or unimportant features. White bits will be placed at locations of the original image that had important bits. This means that, regardless of their original value, all the bits connected to the RB having $V_{\mathrm{RB}j} \geq V_{\mathrm{AVG}}$ will be represented by the white bits or by the output voltage signal $V_{fj} = 1.8$ V. On the other hand, black bits will be placed at locations where the original image had unimportant bits. This means that the remaining bits will be represented by black bits or by the output voltage signal $V_{fj} = 0$ V.

The proposed design was verified using the memristor model proposed by [21] for each synaptic connection. The model parameters were set to emulate the memristor device proposed by [22]. The threshold comparison block was simulated using the BSIM model for a TSMC CMOS technology size of 180 nm. In particular, the comparator consists of pMOS transistors having width-to-length ratios of $W_1/L_1 = W_2/L_2 = 3.24\mu/0.18\mu$ as well as nMOS transistors having width-to-length ratios of $W_3/L_3 = W_4/L_4 = 2.0\mu/0.18\mu$ and $W_5/L_5 = W_6/L_6 = 1.08\mu/0.18\mu$. The inverter, in turn, consists of a pMOS transistor with $W_7/L_7 = 0.72\mu/0.18\mu$ and an nMOS transistor with $W_8/L_8 = 0.36\mu/0.18\mu$. The supply voltages $V_{\mathrm{DD1}} = V_{\mathrm{DD2}} = 1.8$ V.

3) Selection of Parameters and Design Tradeoffs: There are certain circuit parameters that need to be selected according to design tradeoffs, as will be explained in this section. Specifically, this includes the selection of N and M circuit parameters of a single processing block, which is illustrated in Fig. 4, as well as the total number of such processing blocks, indicated by the parameter P, required to create a whole SP.

Fig. 6 shows an exemplary image consisting of $16 \times 16 = 256$ bits along with four different possibilities of how it can be processed. Depending on the selected pair of parameters (N, M), the image can be processed by dividing its bits (green boxes) into RB regions (black boxes) and inhibition regions (red boxes) of various sizes. Although a single RB region with dimensions of $n \times n$ bits reads in total $N = n \times n$ bits, a single inhibition region with dimensions of $m \times m$ incorporates signals from, in total, $M = m \times m$ RBs. As a single processing block processes the input bits of a single inhibition region, the circuit illustrated in Fig. 4 is capable of processing a total of $N \times M$ bits.

The whole SP then consists of P number of such processing blocks. As each processing block is independent in terms of computations, the entire SP is capable of processing $P \times M \times N$ bits simultaneously. This, in turn, allows the whole image to be processed in a parallel manner by dividing its bits into inhibition regions, each of which is processed by a separate processing block. If the input to the system is preprocessed to have fixed dimensions of $X \times Y$ bits, the require parameter P can then be calculated as $P = (X \times Y)/(N \times M)$.

The selection of appropriate N, M, and P parameters is crucial to providing the most optimal system characteristics. In particular, the performance of the proposed SP was evaluated in terms of its on-chip area and dissipated power as well as its qualitative effect on the system's processing time and its quantitative effect on the system's recognition accuracy.

Consider the exemplary image having in total $16 \times 16 = 256$ bits (Fig. 6) processed by the SP constructed by processing blocks having $(N, M) = (1, 4)$ parameters (top-left corner) and $(N, M) = (4, 16)$ parameters (bottom-right corner). Assuming that all input images are preprocessed to have exactly the same dimensions, in the case when $(N, M) = (1, 4)$, the SP should consist of $P = 64$ processing blocks, and in the case when $(N, M) = (4, 16)$, the SP should consist of $P = 4$ processing blocks. As the number of components required for the SP is higher in the first case, the on-chip area and the dissipated power are expected to be higher for $P = 64$ in comparison with $P = 4$. The time required to process an input set is, however, the same, as all the bits are fetched in parallel. Finally, it will be shown later that because the inhibition process with $P = 64$ becomes more localized, the accuracy results are higher for $P = 64$ compared to $P = 4$. Hence, this paper incorporates the analysis to determine circuit parameters to achieve optimal system characteristics based on the face database that was selected for verification.

4) Temporal Memory: The hardware implementation of HTM TM consists of a discrete analog memristive memory array and circuit for updating data in the array based on the proposed HTM TM concept. The memory array consists of memristive memory cells, where each cell corresponds to a single image pixel. For an image of size $A \times B$, a memory array consisting of $A \times B$ memory cells is required.

The operating principle of a single memory cell is based on the ability of a memristor to memorize its state and change the resistance according to the applied voltage. Fig. 7 shows the proposed memristive memory cell consisting of three branches. Such a cell requires five input signals, V_{w1}, V_{w2}, V_{w3}, V_r, and V_c, and produces one output signal: V_o. The input signals V_w, V_r, and V_c correspond to the process of writing (storing) the value to the cell, reading stored values and

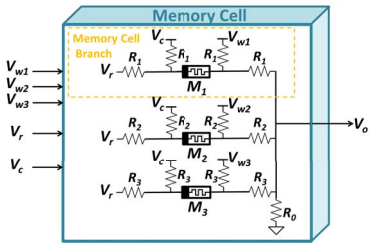

Fig. 7. Proposed memristive memory cell consisting of three branches. V_{w1}, V_{w2}, and V_{w3} are input write voltages; V_r corresponds to the input read voltage; and V_c refers to the input clear voltage. V_o is the output produced during the read cycle. M_1, M_2, and M_3 are memristance values that change according to the applied voltage. R_1, R_2, and R_3 are the resistors values, where $R_1 \neq R_2 \neq R_3$.

clearing the cell, respectively. The memristive memory cell allows one to store $L = v^k$ different values, which implies the use of distinct resistor values ($R_1 \neq R_2 \neq R_3$). The parameter L refers to the number of possible distinct values that can be stored in the cell, v is the number of voltage levels that can be applied to change the memristance of M_1, M_2, and M_3, and k is the number of memory cell branches. For the selected memristor model introduced in [21], three voltage levels v can be applied to change the memristance: 1.2, 2, and 3 V. Fig. 7 illustrates the memory cell with three branches, which means that the number of possible voltage values L that can be stored in the cell is 27. Depending on the application and required memory capacity, the number of cell branches can be increased, which would lead to an increase in L.

To store (write) the value to the memory array, a set of voltages V_{w1}, V_{w2}, V_{w3} must be applied to each cell branch. Each combination of write voltages (1.5, 2, and 3 V) corresponds to a particular voltage value that is stored. During the write cycle, the resistance values of the memristors are changed and preserved until the read cycle. In the write cycle, V_r and V_c are grounded. During the read cycle, the input voltage $V_r = 0.05$ V should be applied, while all V_w and V_c are set to 0. When the read voltage V_r is applied, the cell generates a particular V_o value corresponding to the stored value. To rewrite the data stored in the memory cell, the clear operation is used. To clear the cell, the signal of 3 V should be applied to V_c, while V_r and V_w-s are grounded.

The proposed memristive memory cells form a discrete analog memristive memory array to store the training image of a particular class, where each array cell represents a single image pixel. For c number of classes, c discrete analog memristive memory arrays are required. During the training mode, the memory arrays are updated. Fig. 8 illustrates the circuit diagram of the proposed HTM TM and update operation. The HTM TM update circuit consists of comparator, summing amplifier and thresholding circuit. The first training binary image of each class is saved in a corresponding memory array. During the training mode, each memory array is updated when a new training image of a corresponding class arrives. During the update process, the training image template stored in the memory array becomes grayscale due to the $\pm\Delta$ operation. When the training of a certain class is finished, the final training template is binarized and stored in a corresponding memory array again.

In the training mode, when a new training image from the HTM SP is directed to the HTM TM by the output data controller, each pixel of this image is processed by the comparator. The comparator circuit determines whether a training image pixel is black (with a voltage of 0 V) or white (with a voltage of 1.8 V). If the comparator input V_f is 0, the comparator output V_{cout} is $-\Delta$. If the input is 1, the output becomes $+\Delta$. The comparator consists of two CMOS inverters with pMOS and nMOS transistors. In the first inverter, $V_{\text{DD1}} = 1.8$ V and $V_{\text{SS1}} = -0.5$ V. In the second inverter, $V_{\text{DD2}} = +\Delta$ and $V_{\text{SS1}} = -\Delta$ to ensure that the comparator output V_{cout} is $\pm\Delta$. For this paper, the Δ value is selected as 0.05. For both inverters, the pMOS transistor ratio is $W_1/L_1 = W_2/L_2 = 0.72\mu/0.18\mu$, and the nMOS transistor ratio is $W_3/L_3 = W_4/L_4 = 0.36\mu/0.18\mu$. The second inverter of the comparator circuit is an underdrive inverter to ensure that the $\pm\Delta$ operation can be carried out.

The comparator output is applied to the summing amplifier. The CMOS-memristive summing amplifier consists of an averaging phase, corresponding to the memristors M_1 and M_2, and an amplification phase. The pixel value of the training class map V_t stored in TM and the comparator output V_{cout} are averaged by the memristive averaging circuit with $M_1 = M_2$ to obtain the value of V_{ave}, where $V_{\text{ave}} = (V_{\text{cout}} + V_t)/2$. The memristance values M_1 and M_2 are selected to be approximately 30 $k\Omega$ to eliminate the effect of the summing amplifier on the comparator output V_{cout}. The memristors M_1 and M_2 are preprogrammed to be 30 $k\Omega$ before the training phase. Then, the amplifier doubles V_{ave} to produce an output signal $V_o = V_{\text{cout}} + V_t$, which implies updating of the saved training value V_t by $\pm\Delta$. The modified amplifier configuration proposed in [23] is used in the amplification part. To double the amplifier input V_{ave}, the memristance value ratio $M_4/M_3 = 285k/100k$ is selected. The transistor ratios are $W_5/L_5 = W_6/L_6 = W_8/L_8 = W_9/L_9 = W_{10}/L_{10} = 2\mu/0.18\mu$ and $W_7/L_7 = 2.02\mu/0.18\mu$. The voltage values are $V_{\text{DD}} = 1.8V$, $V_{\text{SS}} = -1.8$ V and $V_B = -0.4$ V. The value of V_{SS} is adjusted to compensate for the effect of the offset in the summing amplifier. In addition, the current source I_c, which provides a current of 36 μA, can be adjusted to ensure a precise $\pm\Delta$ operation and to select the desired level of the output voltage V_o. The output value V_o is within the range from 0 to 1 V.

After the amplification stage, a new training pixel value is either used to update the memory array or is sent to the binarization circuit. If the training image is not the last image for this particular class, the switch S is kept in position 1, and the memory array is updated. During the update process, a memory cell in the array corresponding to this particular pixel is cleared, and the obtained new training pixel is stored (written) to this cell. The voltage level V_o is stored in the memory array using a specific writing circuit and is read from the memory array using the reading circuit. The voltage V_t that is stored in the memory array is within the range from 0 to 1 V. If the training image is the last image for this particular class, the switch is moved to position 2, and the obtained pixel value is binarized using the thresholding circuit. The thresholding circuit generates output $V_{\text{out}} = 1.8$ V if $V_o > V_{\text{th}}$ and $V_{\text{out}} = 0$ V otherwise, where

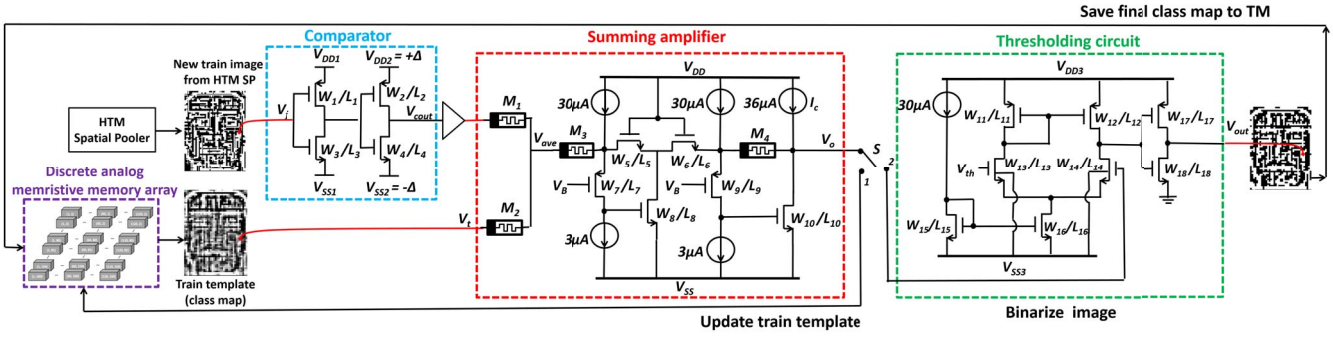

Fig. 8. HTM TM configuration and memory update circuit for a single image class, consisting of a memory array, comparator, summing amplifier, and thresholding circuit. The memristive memory array is used to store the temporary grayscale training class map. When the new training image arrives, the training template is updated using the comparator and the summing amplifier. When the training phase for the class is finished, the training class map is binarized using a thresholding circuit and stored in the same memory array.

V_{th} is the threshold voltage, selected as 0.8 V. The parameters of the thresholding circuit are $V_{\text{DD3}} = 1.8$ V, $V_{\text{SS3}} = 0$ V, $W_{11}/L_{11} = 3.24\mu/0.18\mu$, $W_{12}/L_{12} = 3\mu/0.18\mu$, $W_{13}/L_{13} = W_{14}/L_{14} = 2\mu/0.18\mu$, $W_{15}/L_{15} = W_{16}/L_{16} = 1.08\mu/0.18\mu$, $W_{17}/L_{17} = 0.72\mu/0.18\mu$, and $W_{18}/L_{18} = 0.36\mu/0.18\mu$. The final training pixel output value V_{out} is stored in the same TM array, and the training stage for this particular class is finished. The final class map preserved in the memory array is further used in the recognition (testing) stage. The values that are read from the memory array during the recognition stage are normalized to logic "high" (1.8 V) and logic "low" (0 V), which allows the direct pattern matching operation to be carried out.

5) Memristive Pattern Matcher: During the testing mode of operation of the proposed system, the memristive pattern matcher is used for comparison of the testing image with all class maps. The matcher circuit is based on XNOR threshold logic gates [24] and the averaging operation to produce the output, which demonstrates the weighted value of how many bits from the testing image are similar to the bits of the class map. Fig. 9 demonstrates the circuit diagram of the 2-bit XNOR pattern matcher. As demonstrated in the diagram, the circuit-level implementation of the XNOR gate is based on the XOR gate, which in turn utilizes the memristor-CMOS implementation of the NOR gate.

The memristive NOR gate is constructed by two memristors with memristance values of M_1 and a third memristor having a memristance of M_{NOR} being placed in parallel, followed by the CMOS inverter. The memristance values M_1 and M_{NOR} are selected as high. The same configuration that is used for the NOR gate can be utilized for the NAND gate by controlling the voltage V_c. For a supply voltage level $V_{\text{DD}} = 1$ V, the NOR gate implementation requires a voltage $V_c = 1$, while for the NAND gate, it should be $V_c = 0$. However, for the 180 nm TSMC CMOS process and a supply voltage $V_{\text{DD}} = 1.8$ V, the voltage level V_c and memristor M_{XOR} should be accurately adjusted because of the high sensitivity of these logic gates to the changes in these two parameters. The voltage V_c is set to 0.61 V, and M_{XOR} is preprogrammed such that $M_{\text{XOR}} = 1.2$ MΩ.

The obtained structure of the NOR gate can be used to construct the XOR gate by adding three additional memristors and an inverter. The memristance value M_{XOR} depends on the

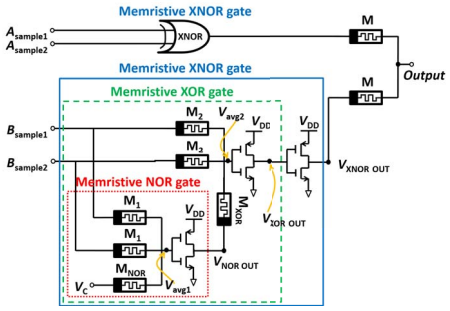

Fig. 9. Two-bit XNOR pattern matcher, created through the combination of memristive XNOR and memristive NOR configurations of memristive memory threshold logic.

output voltage of the NOR gate V_{NOROUT}. For the high voltage value of V_{NOROUT}, which occurs when the two inputs to the gate are low (say, $B_{\text{sample1}} = B_{\text{sample2}} = 0$), M_{XOR} should be taken as a low value. For all the other combinations of two gate inputs, which produce an output voltage of the NOR gate of $V_{\text{NOROUT}} = 0$, the value of the memristance M_{XOR} is set as high. Finally, an additional inverter is used to obtain the XNOR logic gate. All the three inverters are identical, with the voltage supply of $V_{\text{DD}} = 1.8$ V. The parameters for the transistors of the XNOR pattern matcher circuit are as follows: $W_1/L_1 = W_3/L_3 = W_5/L_5 = 0.72\mu/0.18\mu$ and $W_2/L_2 = W_4/L_4 = W_6/L_6 = 0.36\mu/0.18\mu$ for pMOS and nMOS transistors, respectively. The memristors M_1, M_2, and M_{NOR} are set to $R_{\text{off}} = 2.5$ MΩ.

The outputs from the memristive XNOR gate are averaged using the set of memristors with the same high memristance value M. The final *Output* from the memristive pattern matcher circuit can be treated as a weighted similarity score. This is because the obtained value represents the number of testing bits that are similar to the bits in the class map divided by the number of bits. The averaging operation can be replaced by the summing operation, which would show the total number of similar bits; however, this would require significantly larger on-chip area and increased power dissipation. The class number can be found by fetching all pattern matcher outputs and sending them to the WTA circuit and then by obtaining the maximum XNOR output. The maximum XNOR output corresponds to the minimum difference between the images.

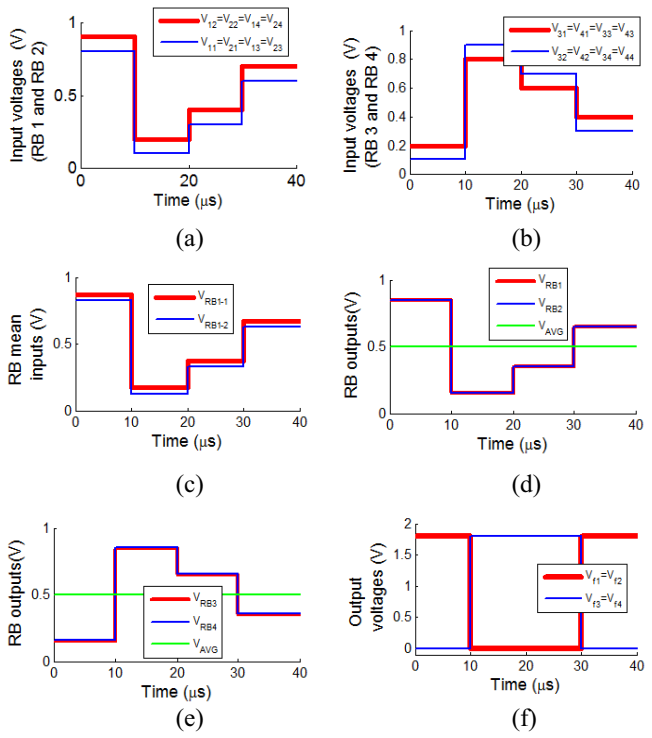

Fig. 10. Timing diagram for HTM SP operation. (a) Input voltages of RB 1 and RB 2, (b) input voltages of RB 3 and RB 4, (c) example of the outputs of the bunches of random weight synapses inside the RB, (d) outputs of RB 1 and RB 2 and threshold calculation V_{AVG}, (e) outputs of RB 3 and RB 4, and (f) HTM SP output pattern.

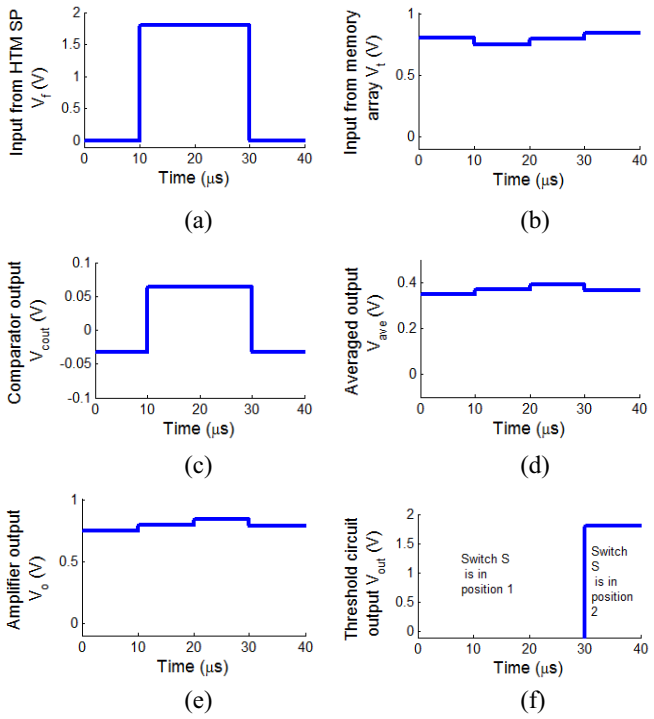

Fig. 11. Simulation results of the HTM TM update circuit, which consists of four clock cycles. In the first three clock cycles (0−30 μs), the switch S is in position 1, whereas in the last clock cycle (30−40 μs) the switch S is in position 2. The input signal V_f refers to the new training image pixel. V_t is the value of the class map currently stored in the memory array. V_{cout} is the comparator output, V_{ave} is the output of the averaging stage of the amplifier, V_o corresponds to the amplifier output, and V_{out} is the output of the thresholding circuit corresponding to the binarized final training class map.

IV. Verification Results of the New Revised Architecture

A. Architecture Performance Results

1) HTM SP Simulation Results: The simulations are performed using 0.18 CMOS TSMC technology and the 50 nm× 50 nm titanium dioxide TiO_2 memristor models in SPICE and VerilogAMS [21]. The SPICE codes required to simulate large circuits are generated using MATLAB scripts. Fig. 10 presents the simulation results of the feature processing circuit for the inhibition block with $M = 4$ and $N = 4$, i.e., the inhibition block containing four RBs with four inputs per block. Fig. 10(a) and (b) presents the pulse waveforms that were generated as the inputs to the RBs in the sample SP configuration, where RB 1 and RB 2 have the same input signals and RB 3 and RB 4 have the same input signals. The same set of voltages was selected to show that the outputs of the RBs for the same inputs are slightly different, as shown in Fig. 10(d) and (e). This discrepancy is caused by the random weights of the memristive synapses. An example of the voltage distribution inside an RB is shown in Fig. 10(c). The inputs of a particular RB are averaged ten times using the memristors, yielding ten different outputs V_{RM_N}. These ten outputs are sent to the RB mean to produce the final RB output V_{RBM}, which corresponds to ten iterations to ensure a sparse random distribution of the input pattern. Even with the same set of input voltages, the separate bunches of random weight synapses (Fig. 6) can produce slightly different outputs after the mean operation due to randomization of the memristive synapses.

The average of the four RB output values is set as the threshold value, which is denoted as V_{AVG}. As the difference between the outputs of the RBs is not large and the inputs are selected symmetrically, the threshold V_{AVG} shown in Fig. 10(d) and (e) is the same for all four clock cycles. The comparator performs the comparison between the obtained threshold value and the RB outputs. The output from the comparator is inverted. As soon as the threshold value is applied to the positive input of the comparator, the inhibition block output is high (1.8 V) if the RB output V_{RBM} is higher than V_{AVG}, where $M = 1, 2, 3, 4$ in this example, and vice-versa. The output voltages of the inhibition block corresponding to the four RBs are shown in Fig. 10(f).

2) HTM TM Simulation Results: Fig. 11 illustrates the simulation results for the HTM TM circuit. The results of four clock cycles are presented. In the first three clock cycles, the train image is not assumed to be the last, and the switch S is set to position 1. The output of the summing amplifier V_o is stored in the memory array. The output from the previous clock cycle becomes the summing amplifier input V_t of the subsequent clock cycle. In the last clock cycle, we assume that the image is the last training image for the particular class, and the switch S is set to position 2. The amplifier output V_o is sent to the thresholding circuit, which produces the output V_{out} for the final training image pixel for this particular class.

Fig. 11(a) shows the input to the comparator V_f from HTM SP, and Fig. 11(b) illustrates the input to the summing

amplifier V_t from the training class map (discrete analog memristive memory array). Fig. 11(c) presents the comparator output V_{cout}, and Fig. 11(d) shows the output of the averaging stage of the amplifier V_{ave}. Fig. 11(e) presents the outputs of the summing amplifier V_o. Fig. 11(f) corresponds to the thresholding circuit output V_{out}. In the first clock cycle (from 0 μs to 10 μs), the input V_f from the new training image is 0 V, and the corresponding comparator output V_{cout} is approximately $-\Delta = -0.05$. The previously stored training class map value V_t is 0.8 V. The average value V_{ave} of V_t and V_{cout} is 0.35 V. The summing amplifier output V_o for this clock cycle is 0.75 V, which means that the initial value stored in the memory array $V_t = 0.8$ V was reduced by 50 mV. The output V_o is stored in the TM array and used as the input V_t of the second clock cycle. Even if the logic high output from HTM SP corresponds to 1.8 V, the logic high output produced by the summing amplifier and stored in the TM array is normalized between 0 V for logic low and 1 V for logic high to ensure the correct addition of the $\pm\Delta$ value. The thresholding circuit is turned off for the first clock cycle.

For the second clock cycle (from 10 μs to 20 μs), the pixel value of the new training image V_f is 1.8 V, corresponding to the comparator output $V_{\text{cout}} = +\Delta = +0.05$. The averaging circuit output $V_{\text{ave}} = 0.37$ V, and the amplifier output $V_o = 0.794$ V, which are stored in the TM array. In the last clock cycle, the switch is set to position 2, which activates the threshold circuit. In the last clock cycle, the summing amplifier output is 0.789 V, and the thresholded output $V_{\text{out}} = 1.8$ V. This output is stored in the TM array and used in the recognition stage.

3) Pattern Matcher Simulation Results: Fig. 12 shows the simulation results for the pattern matcher (Fig. 9). Fig. 12(a) and (b) illustrates two inputs to the XNOR gates. The four clock cycles represent four different combinations of logic high (1.8 V) and logic low (0 V) values. Fig. 12(c) presents the average value V_{avg1} for the memristive NOR gate. Fig. 12(d) presents the output of the memristive XNOR gate V_{NOROUT}. In the ideal NOR gate output, the first three clock cycles must be equal to 0. However, because the circuit was adjusted to make it compatible with the 180 nm CMOS technology and the nominal V_{DD} was set to 1.8 V, the NOR gate output is not precise, which is not critical in this circuit configuration. Fig. 12(e) presents the average voltage V_{avg2} for the memristive XOR gate and the inverter threshold. To obtain nominal values for the 180 nm technology, the inverter threshold was increased to approximately 0.8 V. Fig. 12(f) shows the XOR gate and XNOR gate outputs. The final XNOR output is high for the same inputs and low for different inputs. Finally, the outputs of the pattern matchers are averaged and sent to the WTA. Therefore, if the input pixels of the pattern matcher are the same, the average output value corresponding to a particular class increases.

4) Power and Area Calculations: Table I shows a summary of the area and power calculations for the three circuits that were demonstrated in this paper: SP, TM, and memristive pattern matcher. The values in Table I are represented based on consideration of the minimum on-chip area and the worst-case scenario for power dissipation. The maximum amount of power is dissipated when the inputs to the circuit are supplied by the voltage sources with the maximum value, which is 1 for the applications discussed in this paper, and all random resistance values of the memristors are set to R_{on}.

The area and power for the SP implementation were calculated for different configurations, i.e., different sizes of the receptor and inhibition blocks. A closer look at the area and power calculations for a particular value of M reveals that

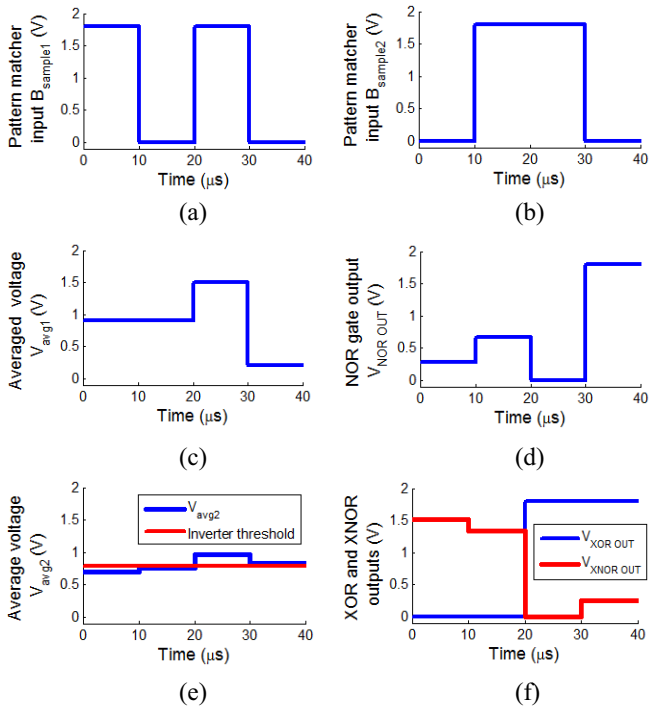

Fig. 12. Memristive pattern matcher timing diagram. (a) Pixel 1 input value B_{sample1}, (b) pixel 2 input value B_{sample2}, (c) average voltage V_{avg1}, (d) NOR gate output V_{NOROUT}, (e) average voltage V_{avg2} and inverter threshold, and (f) XOR and XNOR outputs.

TABLE I
AREA AND POWER CALCULATION FOR THE PROPOSED MODIFIED HTM DESIGN

Configuration		Area (μm^2)	Maximum Power (μW)
HTM SP			
M = 4	N = n×n = 1×1 = 1	19.96	365.88
	N = n×n = 2×2 = 4	20.26	365.88
	N = n×n = 3×3 = 9	20.76	365.88
M = 9	N = 1	44.9	823.23
	N = 4	45.58	823.23
	N = 9	46.70	823.23
M = 16	N = 1	79.83	1463.52
	N = 4	81.03	1463.52
	N = 9	83.03	1463.52
M = 25	N = 1	124.73	2286.75
	N = 4	126.60	2286.75
	N = 9	129.73	2286.75
HTM TM			
$A = 1$, $B = 1$ (for single pixel)		23.85	442.26
Memristive pattern matcher			
2 bit matcher		1.18	69.44

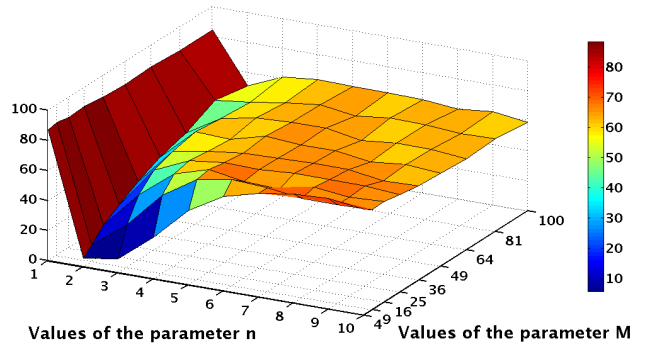

Fig. 13. Two-variable analysis performed to include recognition accuracy as an additional criterion in the selection of optimal values for the $N = n \times n$ and $M = m \times m$ circuit parameters.

values do not change significantly with increasing $n \times n$ size. More precisely, the values of the maximum dissipated power do not change for a given M. This is because an increase in N increases the number of synapses within a single column, which are in turn represented by the memristive devices, which are very compact and do not dissipate significant amounts of power. In contrast, an increase in M results in more columns. A single column require a comparator and an inverter for the processing. Thus, the threshold comparison operation significantly affects on-chip area and power requirements.

Table I also shows that the pattern matching could be implemented by the compact CMOS-memristor-based circuitry at low power consumption.

5) Selection of the Optimal Parameters: In addition to these values, a two-variable analysis was performed to include recognition accuracy as an additional criterion in the selection of optimal values for N and M. This was achieved by performing face recognition analysis on the AR face database [25] with the initial assumption that the optimal delta for TM should be as small as $\Delta = \pm 0.05$. Fig. 13 illustrates the relation between the n and M circuit parameters and the recognition accuracy results for input images having fixed dimensions of 120×160 bits. As can be seen, the best recognition accuracy for the AR database is achieved with the small values of the n and M parameters. This is because when the inhibition region dimensions increase, the number of bits being suppressed by the inhibition block increases as well. This means that small values for n and m must be selected to increase the number of regions within which decisions are made and to ensure that there is no loss of important features.

Considering the results illustrated in Table I, for a fixed input of $120 \times 160 = 19\,200$ bits, the optimal circuit parameters were selected to be $N = n \times n = 1$ and $M = m \times m = 4$, for a total of $P = 4800$ processing blocks, a single unit of which is illustrated in Fig. 4. Thus, for the proposed design, the SP weights are not trained. Rather, learning is initially achieved via standard filtering; subsequently, mainly within the TM part, the optimal performance is achieved when either all or none of the potentials of the SP part are activated simultaneously.

Fig. 14 illustrates the analysis that was performed to determine the optimal delta parameter required for TM. This was achieved with the SP having the fixed parameters listed above and performing face recognition for different values of training

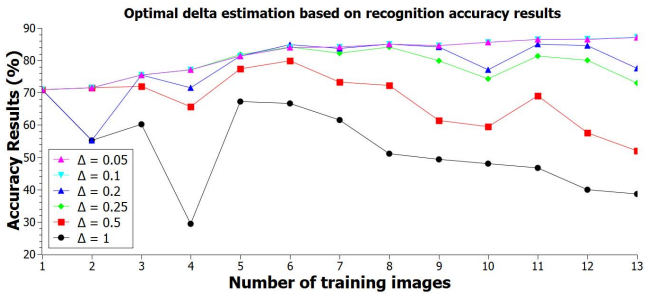

Fig. 14. Optimal Delta estimation based on recognition accuracy results, with the SP having fixed circuit parameters of $N = 1$, $M = 4$, and $P = 4800$.

images. It can be observed that as the number of training images increases, the best recognition accuracy is achieved $\pm\Delta$ with a value of less than or equal to ± 0.1. Hence, the optimal delta value was selected as $\Delta = \pm 0.05$.

B. End-to-End Evaluations for Applications

1) Face Recognition: The algorithm was tested using two human face databases: 1) AR and 2) ORL. The AR database is the largest database having 100 classes of images, meaning that the faces of 100 different people are taken. There are 26 face images per person with different facial expressions, emotions and occlusions such as light, scarves and eye glasses [25]. The other tested database is the ORL database, which includes 40 classes of images containing ten different images [26]. The database contains different facial expressions and occlusion details in addition to rotated faces up to 20 degrees and 10% scale variations [26]. This enables the evaluation of the impact of facial angle and scale changes on face recognition accuracy and on the efficiency of the proposed algorithm. The overall simulation of the system for performance analysis is carried out in MATLAB by utilizing the results of circuit level simulations using the SPICE tool. In all our experiments, the dataset is divided into two distinct sets: 1) training images and 2) testing images. In addition, no overlap is allowed between these sets.

Fig. 15(a) illustrates the recognition accuracy results for the AR face database for various ratios of training-to-testing images, thereby comparing three different architectures. The first architecture is based only on the conventional HTM SP presented in [11]. The second architecture is the conventional HTM SP [11] along with the analog HTM TM presented in this paper. The third architecture is the proposed modified HTM SP with the proposed analog HTM TM.

The effectiveness of including the proposed analog TM can be observed from the more accurate results achieved by the two architectures utilizing it in contrast to the pure conventional HTM SP architecture. Next, a comparison of the accuracy results achieved by the modified HTM SP with those achieved by the two other architectures emphasizes the advantage of changing the decision criterion within the inhibition region.

Similarly, Fig. 15(b) illustrates the same pattern when the same circuit parameters are used for face recognition on the ORL face database for various ratios of training-to-testing images.

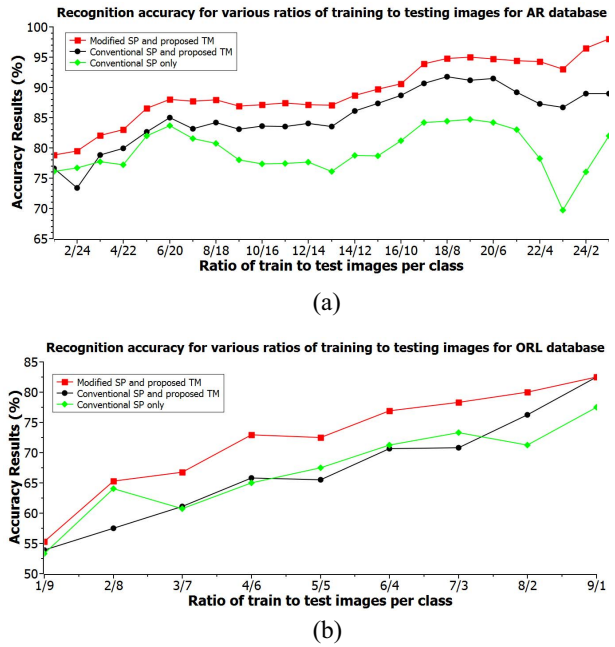

Fig. 15. Recognition accuracy results achieved using three different architectures for various ratios of training images to testing images for (a) AR face database and (b) ORL face database.

TABLE II
RESULTS OF CLASSIFICATION OF THE TEST IMAGES INTO EACH CATEGORY OF THE AR DATABASE UNDER THE THREE DIFFERENT ARCHITECTURES USING A SINGLE TEMPLATE OR A CLASS MAP FOR EACH CLASS, CONSISTING OF 13 TRAINING IMAGES AND 13 TESTING IMAGES

Architecture	Emotions	Light conditions	Occlusions (glasses)	Occlusions (scarf)	**Total**
Conventional HTM SP [11]	77.50%	91.00%	84.33%	53.33%	**76.54%**
Conventional HTM SP [11] with the proposed TM	84.25%	96.33%	85.67%	67.67%	**83.48%**
Proposed (Modified) HTM SP with the proposed TM	85.50%	97.67%	91.33%	74.33%	**87.21%**

Table II illustrates the results when classifying test images of the AR database into one of the four categories under the three different architectures. The setup divided the AR database images into 13 training and 13 testing images for each class. For architectures utilizing the proposed TM, training images were used to create a single class map, which was then used to determine the class of the testing images. For the architecture that is based only on the conventional HTM SP proposed in [11], the training images are initially averaged and only then fetched to the SP to produce a single training template, which was then used to determine the class of the testing images. This setup is implemented to enable fair comparison of the three architectures.

It can be observed that the addition of the proposed analog TM increases the capability of the system to differentiate different categories by almost 7%. Moreover, the change in the decision criterion within the inhibition region, which is the replacement of the conventional SP circuitry by the modified SP circuitry, results in an additional increase in the accuracy of approximately 4%.

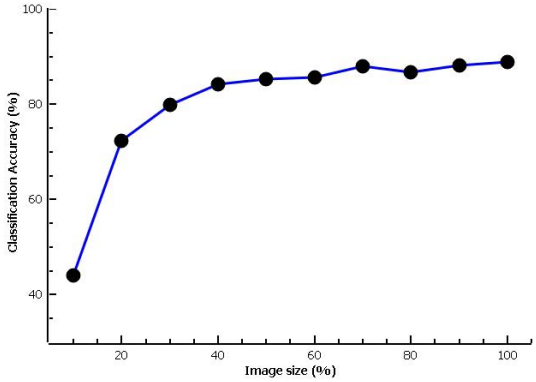

Fig. 16. Impact of image resolution or generally scaling on the classification accuracy. The 100% image size in x-axis corresponds to 120 $\times$ 160 pixels.

Fig. 16 shows the impact of image scaling on the recognition accuracy using the proposed HTM architecture. It can be observed that with increase in resolution there is an increase in the accuracy for the optimal set of parameters.

2) Speech Recognition: The proposed system was also verified on a speech recognition application. The TIMIT database [27] was used to estimate the capability of the proposed system to recognize quickly varying temporal patterns. Although the database consists of a vast number of unique words, few of them, mostly articles and prepositions, are repeated enough to perform training of the system. Hence, only two samples, which are usually used for speaker recognition, are used in this analysis as two separate classes. Two classes are created by combining instances of the *sa1* sample and instances of the *sa2* sample, which are given as

sa1: She had your dark suit in greasy wash water all year.
sa2: Do not ask me to carry an oily rag like that.

Then, the training set is created by combining 50 instances, and the testing set is created by combining 15 instances of each class, providing a total of 130 instances to process by the system.

As the proposed system is constructed to process input data in the form of an image, speech waveforms are initially preprocessed using the perceptual linear prediction (PLP) feature extraction method. Images representing 12th-order PLP features of sample waveforms without RASTA filtering were obtained according to the procedure and codes described in [28]. As a result, speech samples were converted into images without significant degradation of temporal details. These images were then fetched to the proposed system to perform a recognition procedure similar to that used to perform face recognition.

Fig. 17 illustrates the recognition accuracy results for speech recognition with initial waveforms having either no additive white Gaussian noise or AWGN with SNR being equal to 20, 10, or 5 dB. It should be noted that because the PLP feature extraction method produces output images having dimensions of 420×560 bits, which is much larger than the dimensions of the AR images, the M parameter was adjusted accordingly to $M = 49$ RBs within a single inhibition region. With the intent to increase the dimensions within which the decision rule of the Modified SP is performed, this results in an increase in recognition accuracies and in noise robustness.

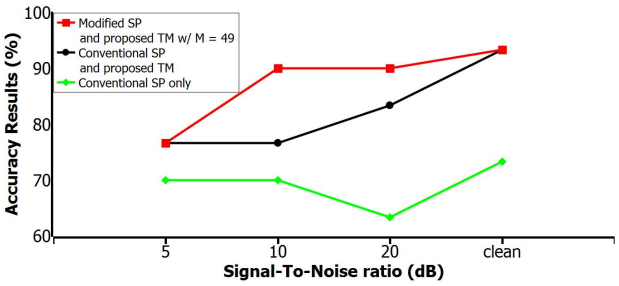

Fig. 17. Speech recognition accuracies obtained under the three different architectures for various SNRs using the PLP feature extraction method as preprocessing.

V. Discussion and Comparison

In this paper, we presented the circuit-level design of an HTM system for pattern recognition that incorporates the optimized architectures of the HTM SP and HTM TM. The simulation results on the optimum values of the M and N parameters for the processing of P blocks for fixed image dimensions revealed that the highest accuracy is achieved with the lowest M and N values. Thus, for image processing on images with dimensions of 120×160, the SP circuit will consist of $P = 4800$ processing blocks, each having $M = 4$ columns with $N = 1$ synaptic connection per column. As a result, using Table I, the total on-chip area and power consumption for the implementation of the proposed SP circuit for image processing are 0.096 mm^2 and 1756 mW, respectively. Compared to the previous works [11] and [19], one of the major advantages of the proposed HTM architecture is the scalability of the design. As the previous implementations of the HTM SP were based on the crossbar architecture, the sneak path problem limited the application of HTM to small-scale tasks. However, the use of memristors and averaging circuits in the proposed HTM design ensures the scalability of the architecture and the application of HTM to large-scale problems. Consideration of the recent publications on the topic of HTM proves the feasibility of the proposed design.

One of the recent designs of the SP was presented in our earlier work [11]. The design [11] is based on parallel memristive crossbar arrays and presents the implementation of the conventional SP algorithm. In this paper, we demonstrated the superiority of the modified HTM over the conventional HTM in terms of achieved recognition and classification accuracy results. In addition to the improved accuracy results that were demonstrated in Section V, the proposed design of the SP circuit offers a reduction in the time required for the processing of an image. This is because the crossbar implementation requires three cycles for operation. The first two cycles are related to the write operation of the memristive crossbar. The two-step write technique is applied to write the high and low values on the memristive devices, which will represent strong and weak synaptic connections. The third cycle is referred to as the read operation, during which the connectedness of the synapse is checked. In the research work [11], the best simulation results were demonstrated for a crossbar design that is based on the 6×6 inhibition block. This implies that for the processing of an input image having $120 \times 160 = 19\,200$ bits, the memristor-crossbar architecture for the SP [11] will be composed of four parallel crossbar slices each having nine synapses (nine rows) within each of 533 serial columns. The sneak path leakage current issue associated with the crossbar structure does not allow for operations within each of the serial columns to be performed in parallel. As a result, considering the switching speed of memristive devices of 10 ns as well as three cycles for the operation of the proposed architecture, the minimum time required for the processing of an image will be $3\ \text{cycles} \times 10\ \text{ns} \times 533\ \text{serialcolumns} = 15\,990\ \text{ns} \approx 16\ \mu\text{s}$. In contrast, the design proposed in this paper allows the processing to be performed in parallel. The delay in the circuit is attributed to the memristor switching time of 10 ns and the amplifier and inverter response times of approximately 2 ns. Thus, the response time of the circuitry will be significantly lower.

Another hardware design of an SP was presented by Streat *et al.* [29]. The proposed nonvolatile architecture [29] was implemented in the VHDL and demonstrated a high level of classification accuracy (91.89%). This design showed requirements in terms of area footprint of 104.26 mm^2 and power consumption for an 8-channel model of the SP of 64.394 mW. The significant reduction in the on-chip area requirement for the implementation of the proposed design can be noticed and is due to the use of nano-scale memristive devices and 180-nm TSMC CMOS technology. However, the use of amplifiers introduces a significant portion of the power dissipation. This problem can be solved by designing a better circuit for the comparison operation. In addition, the increase in the number of synaptic connections N will reduce the total number of amplifiers required for processing and consequently will reduce the power consumption. However, this might lead to a reduction in recognition accuracy because, as mentioned earlier, a large N will reduce the number of regions in which the decision is made upon the importance of the features.

VI. Conclusion

This paper presented several novelties in the area of a brain-inspired machine learning algorithm known as HTM. We proposed a simplified algorithm for SP realization based on averaging operations as well as its possible implementation as the memristor-CMOS circuit. In addition, we reconsidered the concepts of TM and demonstrated an analog memristive memory array for its hardware implementation. We discussed both algorithms and circuit implementations in detail and demonstrated the accuracy and efficiency of the proposed methods for face and speech recognition applications. We achieved an average accuracy of 87.21% for face recognition and 92% for speech recognition. To process images with dimensions of 120×160, the calculated area and power requirements for the proposed HTM SP design implementation are 0.096 mm^2 and 1756 mW, respectively. The proposed HTM TM circuit design for a single pixel requires 23.85 μm^2 of area and 0.442 mW of power.

Acknowledgment

The support of I. Fedorova and A. Irmanova in the preliminary review of the literature and cross-verification of memory cell is acknowledged.

REFERENCES

[1] D. George and J. Hawkins, "A hierarchical Bayesian model of invariant pattern recognition in the visual cortex," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3. Montreal, QC, Canada, Jul./Aug. 2005, pp. 1812–1817.

[2] "Hierarchical temporal memory including HTM cortical learning algorithms," Numenta, Inc., Redwood City, CA, USA, Tech. Rep. VERSION 0.2.1, 2006.

[3] O. Krestinskaya and A. P. James, "Bioinspired memory model for HTM face recognition," in *Proc. Int. Conf. Adv. Comput. Commun. Informat. (ICACCI)*, Jaipur, India, Sep. 2016, pp. 1528–1532.

[4] N. Farahmand, M. H. Dezfoulian, H. GhiasiRad, A. Mokhtari, and A. Nouri, "Online temporal pattern learning," in *Proc. Int. Joint Conf. Neural Netw.*, Atlanta, GA, USA, Jun. 2009, pp. 797–802.

[5] A. B. Csapo, P. Baranyi, and D. Tikk, "Object categorization using VFA-generated nodemaps and hierarchical temporal memories," in *Proc. IEEE Int. Conf. Comput. Cybern. (ICCC)*, Gammarth, Tunisia, Oct. 2007, pp. 257–262.

[6] I. Ramli and C. Ortega-Sanchez, "Pattern recognition using hierarchical concatenation," in *Proc. Int. Conf. Comput. Control Inf. Appl. (IC3INA)*, Bandung, Indonesia, Oct. 2015, pp. 109–113.

[7] W. J. C. Melis, S. Chizuwa, and M. Kameyama, "Evaluation of the hierarchical temporal memory as soft computing platform and its VLSI architecture," in *Proc. 39th Int. Symp. Multiple Valued Logic*, Naha, Japan, May 2009, pp. 233–238.

[8] L. Rodriguez-Cobo, P. B. Garcia-Allende, A. Cobo, J. M. Lopez-Higuera, and O. M. Conde, "Raw material classification by means of hyperspectral imaging and hierarchical temporal memories," *IEEE Sensors J.*, vol. 12, no. 9, pp. 2767–2775, Sep. 2012.

[9] X. Chen, W. Wang, and W. Li, "An overview of hierarchical temporal memory: A new neocortex algorithm," in *Proc. Int. Conf. Model. Identification Control*, Wuhan, China, Jun. 2012, pp. 1004–1010.

[10] A. M. Zyarah and D. Kudithipudi, "Reconfigurable hardware architecture of the spatial Pooler for hierarchical temporal memory," in *Proc. 28th IEEE Int. Syst. Chip Conf. (SOCC)*, Beijing, China, Sep. 2015, pp. 143–153.

[11] A. P. James, I. Fedorova, T. Ibrayev, and D. Kudithipudi, "HTM spatial Pooler with memristor crossbar circuits for sparse biometric recognition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 3, pp. 640–651, Jun. 2017.

[12] E. C. Gangl, "Evolution from analog to digital integration in aircraft avionics—A time of transition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 3, pp. 1163–1170, Jul. 2006.

[13] J. Hawkins and S. Blakeslee, *On Intelligence*. New York, NY, USA: St. Martin's Griffin, 2004, pp. 156–158.

[14] Y. Cui, S. Ahmad, and J. Hawkins, "The HTM spatial Pooler: A neocortical algorithm for online sparse distributed coding," *bioRxiv*, 2017, Art. no. 085035.

[15] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *Neural Comput.*, vol. 28, no. 11, pp. 2474–2504, 2016.

[16] D. George and J. Hawkins, "Hierarchical temporal memory: Concepts, theory and terminology," Numenta, Inc., Redwood City, CA, USA, Tech. Rep., 2006.

[17] M. Deshpande, "FPGA implementation and acceleration of building blocks for biologically inspired computational models," M.S. thesis, Dept. Elect. Comput. Eng., Portland State Univ., Portland, OR, USA, 2011.

[18] T. Ibrayev, A. P. James, C. Merkel, and D. Kudithipudi, "A design of HTM spatial Pooler for face recognition using memristor-CMOS hybrid circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 1254–1257.

[19] D. Fan, M. Sharad, A. Sengupta, and K. Roy, "Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1907–1919, Sep. 2016.

[20] A. M. Zyarah, "Design and analysis of a reconfigurable hierarchical temporal memory architecture," M.S. thesis, Rochester Inst. Technol., Rochester, NY, USA, 2015.

[21] D. Biolek, Z. Kolka, V. Biolkova, and Z. Biolek, "Memristor models for SPICE simulation of extremely large memristive networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 389–392.

[22] M. D. Pickett *et al.*, "Switching dynamics in titanium dioxide memristive devices," *J. Appl. Phys.*, vol. 106, no. 7, 2009, Art. no. 074508.

[23] H. Sato and S. Takagi, "Low-voltage amplifier with improved linearity using triode region MOSFET," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 2469–2472.

[24] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1734–1746, Aug. 2017.

[25] A. Martınez and R. Benavente, "The ar face database," Centre de Visió per Computador, Universitat Autònoma de Barcelona, Bellaterra, Spain, Tech. Rep. 24, 1998.

[26] R. Ahdid, S. Safi, and B. Manaut, "Approach of facial surfaces by contour," in *Proc. Int. Conf. Multimedia Comput. Syst. (ICMCS)*, Marrakesh, Morocco, Apr. 2014, pp. 465–468.

[27] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," NASA, Washington, DC, USA, NASA STI/Recon Tech. Rep. 93, 1993.

[28] D. P. W. Ellis. (2005). *PLP and RASTA (and MFCC, and Inversion) in MATLAB*. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[29] L. Streat, D. Kudithipudi, and K. Gomez, "Non-volatile hierarchical temporal memory: Hardware for spatial pooling," *arXiv preprint arXiv:1611.02792*, 2016.

Olga Krestinskaya (GS'16) received the Bachelor of Engineering (with Hons.) degree in electrical engineering with a focus on bio-inspired memory arrays in 2016. She is currently pursuing the graduation degree in neuromorphic memristive system with Electrical Engineering Department, Nazarbayev University, Astana, Kazakhstan.

Her current research interests include hierarchical temporal memory and pattern recognition algorithms.

Timur Ibrayev (S'16) received the Bachelor of Engineering (with Hons.) degree in electrical engineering with a focus on HTM circuits in 2017. He is currently pursuing the Ph.D. degree with Purdue University, West Lafayette, IN, USA.

His current research interests include memristive HTM circuits and systems for neuromorphic vision, pattern recognition system, and pattern recognition circuits using HTM.

Dr. Ibrayev was a recipient of the Travel Grant for presenting a paper in ISCAS 2016.

Alex Pappachen James (SM'13) received the Ph.D. degree from the Griffith School of Engineering, Griffith University, Nathan, QLD, Australia.

He is currently the Chair of Electrical Engineering Department, Nazarbayev University, Astana, Kazakhstan. His current research interests include brain-inspired circuits, memristor circuits, algorithms and systems. He has a sustained experience of managing industry and academic projects in board design, very large scale integration and pattern recognition algorithms, and semiconductor industry.

Dr. James is an Associate Editor of *Human-Centric Computing and Information Sciences*, IEEE ACCESS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE (special issue), and *IET Cyber-Physical Systems: Theory and Applications* (special issue). He is the Chair of IEEE Kazakhstan section. He is a Senior Fellow of Higher Education Academy, U.K.