IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2019, accepted November 29, 2019, date of publication December 10, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958895

Double Layers Self-Organized Spiking Neural P Systems With Anti-Spikes for Fingerprint Recognition

TONGMAO MA[1,2], SHAOHUA HAO[1], XUN WANG[1], ALFONSO RODRÍGUEZ-PATÓN[2], SHUDONG WANG[1], AND TAO SONG[1,2]

[1]College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China
[2]Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid (UPM), 28660 Madrid, Spain

Corresponding authors: Xun Wang (wangsyun@upc.edu.cn) and Tao Song (tsong@upc.edu.cn)

This work was supported by the Program of China Scholarships Council under Grant 201806450042. The work of T. Song was supported in part by the Tai Shan Scholar Foundation, in part by the National Natural Science Foundation of China under Grant 61873280, Grant 61672033, and Grant 61672248, in part by the Key Research and Development Program of Shandong Province under Grant 2017GGX10147, in part by the Natural Science Foundation of Shandong Province under Grant ZR2017MF004, in part by the Fundamental Research Funds for the Central Universities under Grant 18CX02152A and Grant 19CX05003A-6, in part by MINECO AEI/FEDER, Spain-EU under Project TIN2016-81079-R, in part by the InGEMICS-CM Project (FSE/FEDER, Comunidad de Madrid-EU) under Grant S2017/BMD-3691, and in part by the Talento-Comunidad de Madrid under Grant 2016-T2/TIC-2024.

ABSTRACT In this paper, we design a double layers self-organized spiking neural P system with anti-spikes for fingerprint recognition. The system can self-adaptively create and delete synapse between the neurons in different layers and recognize fingerprints by the spike trains emitted out of the output neurons. Data experiments are conducted on FVC2002 and FVC2004 Databases with EER (Equal Error Rate) 9.5% around, and much less parameters are involved in our SN P systems than Capsule Neural Networks. To our best knowledge, it is the first attempt of using SN P systems to do fingerprint recognition, which can also provide theoretical models for spiking neural circuits recognizing fingerprints.

INDEX TERMS Fingerprint recognition, membrane computing, self-organization, spiking neural P systems.

I. INTRODUCTION

Bio-inspired computing, short for biologically inspired computing, is a major subfield of natural computation, whose aim is to abstract computing ideas from biological systems to construct powerful computing models and algorithms. Membrane computing is a new branch of bio-inspired computing, which seeks to discover new computational models from the study of biological cells, particularly of the cellular membranes. The obtained models are distributed and parallel computing devices, usually called P systems.

Human brain is known the most powerful "computing models" in nature, which has abundant computation intelligences built-up from millions of years' evolution. These information processing intelligences have provided ideas to computer scientists in constructing powerful neural-like computing models see e.g. [1]–[4]. Over the last decades, neural-like computing models gain their popularity for their learning capability [5]–[7] and applications in solving real-life problems [8]–[10].

In 2006, under the framework of membrane computing, spiking neural P systems, (namely SN P systems), were proposed by modelling the way neurons communicate via electrical impulses (spikes) [11]. Instead of using sigmoid functions to imitate biological neuron's spiking in artificial neural networks, spiking rules, denoted in the form of production in grammar of formal languages, are used to describe the neuron's spiking behaviors in SN P systems.

It is formulated in [12] that candidates in the third generation of neural networks have some common features.

(1) The concept of time is incorporated into neuron's spiking.

(2) Neurons do not fire at each propagation cycle, but rather fire only when a membrane potential or the stack of spikes reaches a specific value.

(3) Various coding methods exist for interpreting the outgoing spike train as a real-value number, either relying on the frequency of spikes, or the timing between spikes, to encode information.

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Munir.

In terms of features of models, SN P systems fall into the third generation of neural network models. The third generation Spiking Neural Networks (SNNs) comprising of spiking neurons, where information transfer models the information transfer in biological neurons such as the precise timing of spikes or a sequence of spikes [34].

Thereafter, both bio-inspired features and mathematical strategies have been introduced to achieve new variants of SN P systems, such as SN P systems with anti-spikes or inhibitory synapses [13]–[16], SN P systems working in the asynchronous manner [17], with multiple channels [18], Dynamic threshold neural P systems [19], Coupled neural P systems [20], SN P systems with astrocyte-like control [21]–[23], homogeneous SN P systems [24], [25], SN P systems with weight [26], sequential SN P systems [15], [27], [28], SN P systems with exhaustive use of rules [29], [30], SN P systems with rules on synapses [31], [32]. It was formulated in [33] that research on new variants of SN P systems becomes "a promising branch in membrane computing".

It is a natural extension, but still remains an open research problem, to use SN P systems to do image recognition [33]. Recently, it is reported that SN P systems are used to recognize English letters and symbols [34].

In this paper, we use SN P systems for fingerprint recognition. Specifically, a double layers self-organized spiking neural P system with anti-spikes is designed. In the system, a set of template fingerprints are stored in form of a number of spikes in each neuron at the bottom layer, and in the upper layer, the fingerprint to be recognized is stored by a stack of spikes. By creating and deleting synapses between the neurons from different layers, our system can recognize fingerprints by the spike trains emitted out of the output neurons.

Data experiments are conducted on FVC2002 and FVC 2004 Databases with EER (Equal Error Rate) 9.5% around. The accuracy cannot overcome the deep models, but it is acceptable, since much less parameters are involved in our SN P system. Partially, it needs about 50 million for convolutional neural networks and 300 million for capsule neural networks to recognize fingerprints, but the parameters in SN P systems is much less. More importantly the results show the feasibility of recognizing fingerprints by self-organized SN P systems. To our best knowledge, it is the first attempt of using SN P systems to do fingerprint recognition, which can also provide theoretical models for spiking neural circuits recognizing fingerprints.

II. DOUBLE LAYERS SELF-ORGANIZED SPIKING NEURAL P SYSTEMS

It starts by recalling the formal definition of classical SN P systems from [11]. After that, the newly designed double layers self-organized SN P system with anti-spikes is introduced.

A. SPIKING NEURAL P SYSTEMS

An SN P system of degree m is formally defined as a construct of the form:

$$\Pi = (O, \sigma_1, \sigma_2, \ldots, \sigma_m, syn, \sigma_{in}, \sigma_{out}), \text{ where}$$

- $O = \{a\}$ is a singleton alphabet and a is called spike;
- $\sigma_1, \sigma_2, \ldots, \sigma_m$ are neurons of the form $\sigma_i = (n_i, R_i)$, with $1 \leq i \leq m$, where n_i is initial number of spikes in neuron σ_i and R_i is the set of rules in neuron σ_i:

1. Spiking rule : $E/a^c \rightarrow a^p; d$, where E is a regular expression over O,$c \geq p \geq 1$ and $d \geq 0$;

2. Forgetting rule:$a^s \rightarrow \lambda$, for some $s \geq 1$, with the restriction that $a^s \notin L(E)$ for any rule $E/a^c \rightarrow a^p;d$ from any R_i.

- syn is the set of synapses; each element in syn is a pair of the form (i, j), where (i, j) indicates that there is a synapse connecting neurons σ_i and σ_j, with $i, j \in \{1, 2, \ldots, m\}$, $i \neq j$;
- σ_{in} is input neuron and σ_{out} is output neuron.

A spiking rule $E/a^c \rightarrow a^p; d$ is applied as follows. At certain moment t, if neuron σ_i contains k spikes with $a^k \in L(E), k > c$, and rule $E/a^c \rightarrow a^p; d$, it fires by consuming c spikes ($k - c$ spikes remaining in neuron σ_i) and sending one spike to each of its neighboring neurons after d time units, during which it cannot receive new spikes. As for a forgetting rule $a^s \rightarrow \lambda$, if neuron σ_i contains s spikes and $a^s \notin L(E)$, the forgetting rule will be applied. It will consume s spikes and send them to the environment. As a result, the number in neuron σ_i will be 0.

B. DOUBLE LAYERS SELF-ORGANIZED SPIKING NEURAL P SYSTEMS USING ANTI-SPIKES

Considering the comparison between template fingerprint and sample fingerprint, we introduce anti-spikes proposed in [13]–[16] to distinguish them and each of them is represented by one layer. Synapse creating and deleting rules proposed in [35], [36] are used to achieve self-organization function.

A double layers self-organized SN P system with anti-spikes of degree 2m is formally defined as a construct:

$$\Pi_l = \left(O, \sigma_{l1}, \sigma_{l2}, \ldots, \sigma_{lm}, syn_0, in_l, out_l\right), \text{ where}$$

- O $= \{a, -a\}$ is alphabet and a is called spike, $-a$ is called anti-spikes;
- $l = 1, 2$ denotes the label of layers;
- $\sigma_{l1}, \sigma_{l2}, \ldots, \sigma_{lm}$ are neurons of the form $\sigma_{li} = (n_{li}, R_{li})$, with $1 \leq i \leq m$, where n_{li} is initial number of spikes in neuron σ_{li} and R_{li} is the set of rules in neuron σ_{li}, in particular, rules are only used in the same layer:

1. Spiking rule: $E/a^c \rightarrow a^p;d$, where E is a regular expression over O, $c \geq p \geq 1$ and $d \geq 0$;

2. Synapse creating rule: $E'/a^{c'} \rightarrow +\left(a^{p'}, cre(li)\right)$, where E' is a regular expression over O, $cre(li) \subseteq \{\sigma_{l1}, \sigma_{l2}, \ldots, \sigma_{lm}\} / \{\sigma_{li}\}$, and $c' \geq p' \geq 1$;

3. Synapse deleting rule:$E''/a^{c''} \rightarrow -(\lambda, del(li))$, where E'' is a regular expression over O, $del(li) \subseteq \{\sigma_{l1}, \sigma_{l2}, \ldots, \sigma_{lm}\}/\{\sigma_{li}\}$, and $c'' \geq 1$

• $syn_0 = \emptyset$ is the initial set of synapses, it represents there is no synapse at first; at any time, the synapses set can be represented by $syn_t \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$

• σ_{in_l} are input neurons and σ_{out_l} are output neurons, particularly, each layer has different input and output neurons.

A spiking rule of the form $E/a^c \rightarrow a^p;d$ is applied as follows. If neuron σ_{li} contains k spikes, and $a^k \in L(E), k \geq c$, then rule $E/a^c \rightarrow a^p;d \in R_{li}$ can be applied. It means that c spikes are consumed and removed from neuron σ_{li}, i.e., $k-c$ spikes are remained, while the neuron emits p spikes to its neighboring neurons after d steps. (It is a common practice in membrane computing to have a global clock defined. The block is used to mark the time of the whole system and ensure the system synchronization.) If $d = 0$, p spikes are emitted out immediately; if $d = 1$, p spikes are emitted out at next step, etc. Assume that the rule is used in step t and $d \geq 1$, then in steps $t, t+1, \ldots, t+d-1$ the neuron is in close status (this corresponds to the refractory period from neurobiology), and cannot receive new spikes (if a neuron tries to send spikes to a neuron in close status, then these particular spikes will be lost). In the step $t+d$, the neuron fires and regains open status, so it can receive spikes (which can be used starting with the step $t+d+1$, when the neuron can again apply rules). It is possible that p is associated with value 0. In this case, neuron σ_{li} consumes c spikes without emitting any spike. Spiking rule with $p = 0$ is also called forgetting rule, by which a pre-defined number of spikes can be removed out of the neuron. If $E = a^c$, then the rule can be written in the simplified form $a^c \rightarrow a^p;d$, and if $d = 0$, it could be $a^c \rightarrow a^p$.

Synapse creating and deleting rules mainly control synapses' creation or deletion during the computation. Synapse creating rule $E'/a^{c'} \rightarrow +\left(a^{p'}, cre(li)\right)$ is applied as follows. If neuron σ_{li} contains k' spikes such that $a^{k'} \in L(E')$, $k' \geq c'$, then the synapse creating rule is applied with consuming spikes, creating synapses to connect neuron σ_{li} to each neuron in $cre(li)$ and emitting p' spikes to each neuron in $cre(li)$. If neuron σ_{li} contains k'' spikes such that $a^{k''} \in L(E'')$, $k'' \geq c''$, then the synapse deleting rule $E''/a^{c''} \rightarrow -(\lambda, del(li))$ is applied, removing c'' spikes, from neuron σ_{li} and deleting all the synapses connecting neuron σ_{li} to the neurons from $del(li)$.

With the synapse creating and deleting rules, E' and E'' are regular expressions over alphabet $O = \{a, -a\}$, controlling the application of synapse creating and deleting rules. These rules only and if only can be used in some particular spike numbers, which means the neurons are in specific states. Consequently, the system can dynamically rebuild its topological structure during the computation, which is viewed as self-organization. And considering the two layers and anti-spike, it is called double layers self-organized spiking neural P systems with anti-spikes.

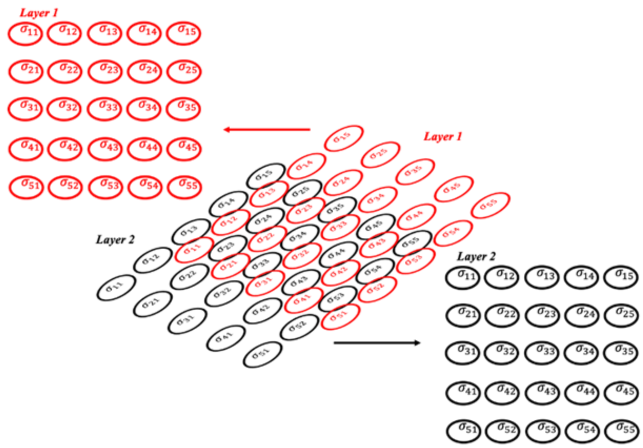


FIGURE 1. An example of double layers self-organized spiking neural p systems with anti-spikes.

P1	P2	P3
P8	P0	P4
P7	P6	P5

FIGURE 2. The pixel model.

As shown in Fig. 1, there is an example of double layers self-organized spiking neural P systems with anti-spikes. Each layer includes 25 neurons with spikes in layer 1 and anti-spikes in layer 2. There is no synapse at first. As described, there are rules managing spikes' creation or delete and synapses' creation and delete as well.

III. FINGERPRINT RECOGNITION BY SN P SYSTEMS

A. IMAGE PRE-PROCESSING

The data structure of SN P systems is spike trains. It needs to do some pre-processing to the fingerprints images, like binarization processing, skeletonization processing and minutiae extraction.

Binarization is to convert gray image to binary image for skeletonization and minutiae extraction. According to characteristics of fingerprint ridges, figure out direction information and make sure the fingerprint ridge not changed.

For any pixel P of certain fingerprint, it is set itself to be the central pixel, and establish a window sized n × n. And as shown in Fig. 2, we give the gray value S_i of pixel P_i with $i = 0, 1, \ldots, 8$. After that, it needs to find out the maximum and minimum gray value of these pixels and check if the binarization formula as follows holds or not,

$$\mathrm{P}+\frac{S_{max}+S_{min}}{2} < \frac{2}{9}\sum_{i=0}^{8} S_i \tag{1}$$

where the S_{max} is maximum pixel, S_{min} is minimum pixel. If this formula is workable, it means this pixel P is a point in ridge. The maximum and minimum points' gray value is less than the mean gray value, so the pixel P is likely in the ridge. Otherwise, it is not a point in the ridge.

We choose Guo and Hall skeletonizing algorithm implementing by SN P systems from [35] to remove some pixels without losing the feature of fingerprints images.

For instance, give a 3×3 pixel model (see Fig. 2), for every pixel P_i, with $i \in \{0, 1, 2, \ldots, 8\}$ is with value 1 (in black) or 0 (in white). We calculate the values of two parameters B (P0) and C(P0) to decide which pixel should be removed.

$$\mathrm{B}\,(P0) = \sum_{i=1}^{8} P_i \tag{2}$$

$$\begin{aligned}\mathrm{C}\,(P0) = {} & (\neg P2 \wedge (P3 \vee P4)) + (\neg P4 \wedge (P5 \vee P6)) \\ & + (\neg P6 \wedge (P7 \vee P8)) + (\neg P8 \wedge (P1 \vee P2))\end{aligned} \tag{3}$$

where B $(P0)$ is the number of black pixel around $P0$, while C $(P0)$ is the connection degree of $P0$. It is obvious that a black pixel's connection degree is 0 if it is an alone black pixel. When there are 8 black pixels around 1 black pixel, its connection degree is 4. As Guo and Hall algorithm suggested, a black pixel P0 can be removed only if it satisfies following conditions.

B (P0) > 1;

C (P0) = 1; make sure local connection is necessary;

(P1 ∧ P3 ∧ P5 ∧ P7) ∨ (P2 ∧ P4 ∧ P6 ∧ P8) = FALES; and obviously, only and only if P0 is not the center of black pixel across, this condition will be satisfied.

It is shown in Fig. 3 a fingerprint after Skeletonization.

Minutiae features such as ridge ending and bifurcations are extracted after pre-processing. Usually, 80 minutiae can be extracted from one fingerprint image. However, there are some fake minutiae because fingerprint images may contain various noises, false traces, blurred ridges and indistinct boundaries. Usually, these fake minutiae need to be removed after extracting. After fake minutiae deletion, we choose about 40-50 feature points left, which will be used to match.

FIGURE 3. Fingerprint and pictures after skeletonization.

B. MINUTIAE MODEL

The traditional method to represent minutiae is very complex and too many information should be used. Fortunately, SN P systems can visual and vivid represent this kind of model.

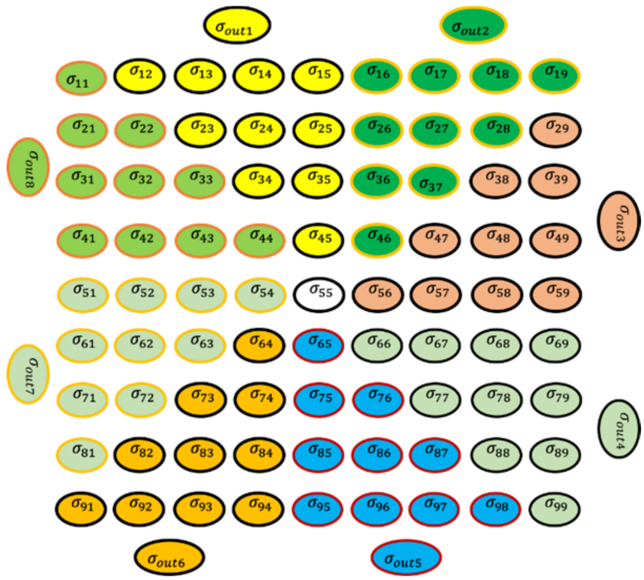


FIGURE 4. The spiking neural p systems model for minutiae.

Based on the SN P systems we proposed, we build models for every minutiae.

For a specific point, set itself as the central and choose a m × m square, where m is adjustable and we choose 9 in this paper. Specially, m can influence the expression of minutiae and results cause it decides the quantity of information around point M. See the model in Fig. 5.

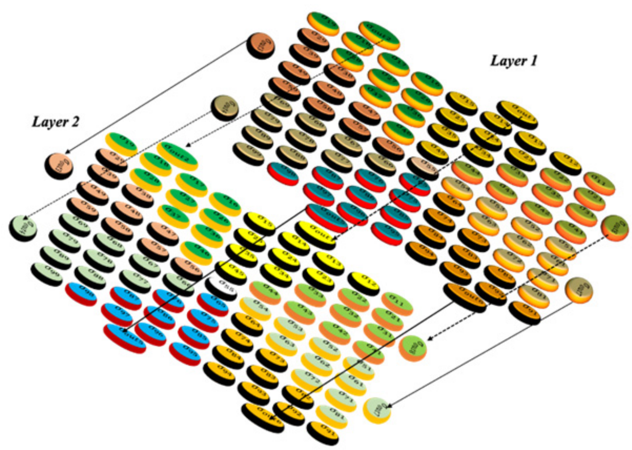


FIGURE 5. Double layers model of the spiking neural p systems.

The model can be formally represented by

$\Pi = (O, \sigma_{11}, \sigma_{12}, \ldots \sigma_{19}, , \sigma_{21}, \sigma_{22}, \ldots, \sigma_{99}, syn_0, out_l)$, where

- $O = \{a\}$ is alphabet and a is called spike, representing the pixel is 1 or black;
- $\sigma_{11}, \sigma_{12}, \ldots \sigma_{19}, , \sigma_{21}, \sigma_{22}, \ldots, \sigma_{99}$ are neurons, representing 81 pixels, and $\sigma_{li} = (n_{li}, R_{li})$, where $1 \leq i, j \leq 9$, where $n_{li} \in \mathbb{N}$, is the initial number within σ_{li}, and it can be sent from input module, and R_{li} is the set of rules in neuron σ_{li}, and three kinds of rules in every neuron.

(1) Spiking rule: $a \rightarrow a$;

(2) Synapse creating rule: $a \rightarrow +(a, p * cre(li, jk))$;

(3) Synapse deleting rule: $\lambda \rightarrow -(\lambda, del(li, jk))$, as long as there is no spike in neuron, this rule will be executed.

• $syn_0 = \emptyset$ is initial set of synapse, and at any time t, $syn_t \subseteq \{1, 2, \ldots, \mathrm{m}\} \times \{1, 2, \ldots, \mathrm{m}\}$.

• out_i is output neurons, where $i \in \{1, 2, \ldots, 8\}$.

In our model, the square is segmented into 8 parts, and the synapse only exists in every same part. Neuron σ_{55} is the very central neuron, which connects with 8 semi-central neurons $\sigma_{44}, \sigma_{45}, \sigma_{46}, \sigma_{54}, \sigma_{56}, \sigma_{64}, \sigma_{65}, \sigma_{66}$. Each semi-central neuron emits its spikes to its output neuron by three gradations along its own path way in the same part. (The neurons in the same color are from the same part). Semi-central neurons $\sigma_{44}, \sigma_{45}, \sigma_{46}, \sigma_{54}, \sigma_{56}, \sigma_{64}, \sigma_{65}, \sigma_{66}$ have p in their synapse creating rule with value 3, it means the rule will be applied three times, each time emitting one spike to the neuron in the outer circle. In the second gradation, the value of p in their synapse creating rule with value 2, and in the third gradation the value of p in their synapse creating rule with value 1. Finally, connections between neuron σ_{li} and output neurons are created with synapse creating rule, where the value of p is 1. In Fig. 6, the connections of neurons in the third part of first layer are shown.

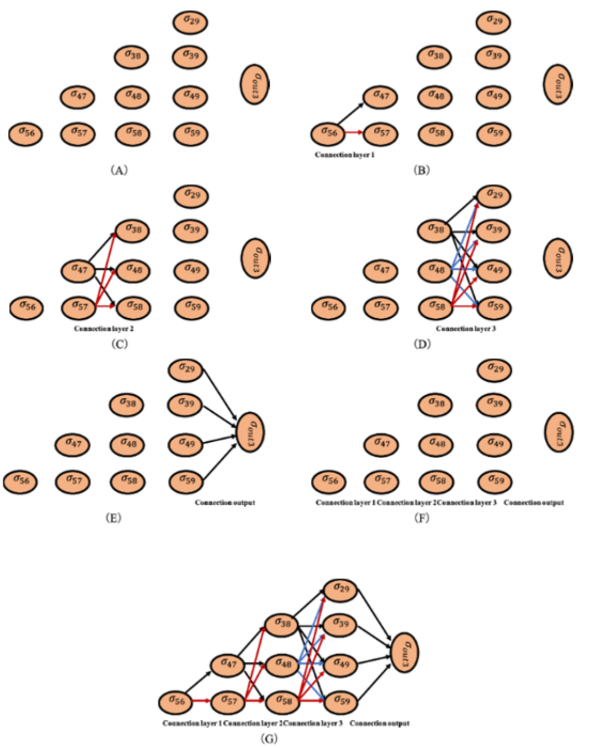


FIGURE 6. The connections of neurons in the thirdpart of first layer.

In Fig. 6, the processing of synapses' creation and delete (connections between neurons) in the third part of first layer are shown. Fig. 6(A) indicates the initial statement of neurons in third part. In Fig. 6(B), synapses are created in connection layer 1, and in Fig. 6(C), synapses are deleted in connection layer 1 and Synapses are created in connection layer 2. In Fig. 6(D), the synapses are deleted in connection layer 2 and Synapses are created in connection layer 3, and in Fig. 6(E), synapses are deleted in connection layer 3 and Synapses are created in connection output. Fig. 6(F) shows synapses are deleted in connection output and the system returns to initial statement, in Fig. 6(G), all the synapses that are created and deleted during the processing. After building a model like this, 8 outputs can represent a minutia.

C. INPUT MODULE

The input module is shown in Fig. 8, which includes one start neuron, one input neuron, $9 \times 9 \times 2$ assistant neurons. The topology of input module is shown in Fig. 7.

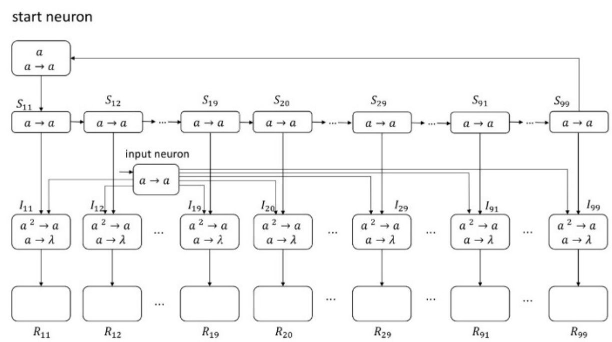


FIGURE 7. Input module of the system.

(A) initial image (B) skeletonization (C) minutiae of template (D) minutiae of sample

FIGURE 8. Images during (a) is the initial fingerprint image and after thinning, we can get picture (b); (c) and (d) are minutiae of template and sample after extraction and fake deletion.

The workflow of input module is as follows: there is only one input neuron that can read information for environment and the binary string $p = p_{11}p_{12} \ldots p_{19}p_{21}p_{22} \ldots p_{91} \ldots p_{99}$ (where $p_{ij} \in \{0, 1\}, i = 1, 2, \ldots, 9, j = 1, 2 \ldots, 9$) is read one by one from left to right. If $p_{11} = 1$, input neuron will receive a spike at t, and this spike if from environment. On the contrary, if $p_{11} = 0$, input neuron won't receive any spike. At step $t = 80$, the input will be finished.

In the initial configuration, there is only one spike in the start neuron while the other neurons don't have spikes. At step t, start neuron send a spike to neuron $\sigma_{S_{11}}$, at the same time, the input neuron starts to read binary string p. As for the input neuron, it has two choices:

(1) if $p_{11} = 1$, input neuron will receive a spike at step t from the environment. In next step, it sends one spike to $\sigma_{I_{11}}$.

Neuron $\sigma_{s_{11}}$ contains one spike at step $t+1$, it fires to send one spike to neuron $\sigma_{I_{11}}$. As a result, neuron $\sigma_{I_{11}}$ accumulates two spikes. In this way, spiking rule $a^2 \rightarrow a$ is enabled to send a spike to neuron $\sigma_{R_{11}}$, which means $p_{11}=1$.

(2) if $p_{11}=0$, input neuron won't receive any spike and the spiking rule within this neuron won't spike. But $\sigma_{s_{11}}$ will send a spike to $\sigma_{S_{11}}$. Then the forgetting rule within $\sigma_{S_{11}}$ will execute. $\sigma_{S_{11}}$ doesn't receive any spike, which means $p_{11}=0$.

At step $t+1$, the second bit p_{12} in binary string can be read by input neuron. If $p_{12}=1$, $\sigma_{R_{12}}$ will contain one spike. Otherwise, no spike exists in $\sigma_{R_{12}}$. Similarly, when reading a string, $p_{ij}=1$ means $\sigma_{R_{ij}}$ contains one spike, and $p_{ij}=0$ means there is no spike, where i = 1,2,...,9, j = 1,2,...,9. And at $t+80$, input module finishes the string reading and every neuron return back to initial statement except $\sigma_{R_{ij}}$. In fact, spike number of $\sigma_{R_{ij}}$ is 0 or 1.

With input module, the minutiae model can get the spikes for each neuron.

D. ALGORITHM

Usually, about 20-30 minutiae are left to match, and the number of minutiae for template and sample points are different. The matching degree for template and sample is denoted by the value of M calculated by

$$M = \frac{2m}{n_1+n_2} \tag{4}$$

where m is the amount of successful matching points, n_1 is minutiae amount of template while n_2 is amount of sample.

Another layer is needed here. One layer is template model which uses anti-spikes, and the other layer is sample model that uses normal spike. And these two layers are connected by synapses between output neurons in each layer. And these are only connection between two layer. These two layers are the same except use different spikes. Output neurons in first layer will send anti-spikes to output neurons in second layer. And a spike plus an anti-spike will be 0. At last, we can get eight outputs which represent matching degree between two minutiae.

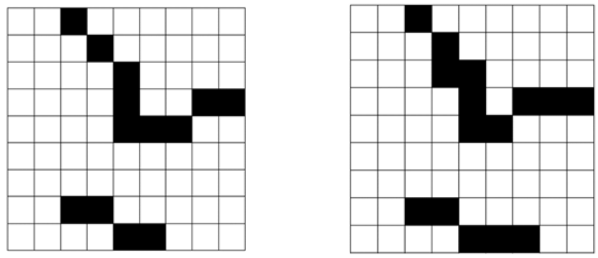
$$M' = \left|\sum_{i=1}^{8} out'_i\right| \tag{5}$$

$out'_i = out^1_i + out^2_i$, if it is bigger, the difference is larger. And the threshold is OUT. If $out'_i < OUT$, these two minutiae are from the same fingerprint, thus m will plus one. Otherwise, they are not matched.

Fingerprint recognition algorithm we proposed is as follows:

(1) Initialization. $m = 0$ (it is a global variable), n_1, n_2, M, i.

(2) According the minutiae amount of template, build n_1 double layers self-organized spiking neural P systems using anti-spikes models $\Pi_1, \Pi_2, \ldots, \Pi_{n_1}$, where first layer uses anti-spikes and send layer uses spikes. And input binary string about template's minutiae into every first layer using anti-spikes.



(a) bifurcation from template (b) bifurcation from sample

FIGURE 9. The minutiae model.

(3) For one of these models Π_i, pick one minutia from sample and input its binary string by input module into second layers.

(4) The SN P systems start running until termination configuration, and the results are 8 variables. Through formula, calculate M' and compare it with M, if it satisfies the condition, $m+1$ and halt this model. Then delete this minutia from sample, so it won't show up in other models either. And turn to step (6). Otherwise, these two are not the same, and turn to step (5).

(5) Considering the angle shifting, change the synapse between output neurons. out_1, out_2, ..., out_8 in first layer will create synapses to out_2, out_3, ..., out_8, out_1. And do the step (3) and (4), if the condition is still not satisfied then considering left shifting, out_1, out_2, ..., out_8 in first layer will create synapses to out_8, out_1, out_2, ..., out_7. And do the step (3) and (4).

(6) Execute step (3)-(5) for all Π until the matching is finished.

(7) Output successful matching points' amount m and calculate the value of matching degree M.

Step (5) is only for angle shifting from -30° to 30°, which may be cased by human activity when fingerprinted.

IV. EXPERIMENTAL RESULTS

We choose databases from Fingerprint Verification Competition 2002 and 2004. There are 80 fingerprint images including 10 fingers and every finger was collected 8 times in each database.

For example, in DB2 of FVC2002, we choose the first image of 10 fingers as the templates and the others are samples. Then choose 103-1 and 103-2 as an example to show the algorithm. As shown in Fig. 9, after pre-processing, minutiae left in template are 44 and left in sample are 53. Then we picked two bifurcations separately from template and sample shown in figure. According bifurcation from template, we input it into the first layer, and input the binary string from sample into the second layer.

The first layer will send anti-spikes (0, −173, 0, −144, 0, −2, −8, 0) into second layer's output neuron, and they are (0, 157, 0, 144, 0, 3, 8, 0). Finally, the system output (0, −24, 0, 0, 0, 1, 0, 0), and the point matching degree M' is 23. The threshold is 144, so we think these two minutiae are the same.

TABLE 1. EER and accuracy on different databases.

Database	Accuracy	EER
DB1(FVC2002)	91.8%	6.7%
DB2(FVC2002)	91.2%	5.8%
DB1(FVC2004)	89.9%	12.5%
DB2(FVC2004)	89.3%	13.2%

After all systems finished, 20 matching points matched, so the matching degree $M = \frac{2\times 20}{44+53} = 0.41$, and according to the threshold, we think these two pictures are fingerprinted from same finger.

Finally, after running all pictures from DB2, the accuracy is 91.8% which is not good compared with most results in the competition, which can achieve 99% accuracy in recognizing fingerprints [37].

What's more, there exits deep learning models with higher accuracy. But it needs 6.58 million parameters for Siamese network, 6.43 million parameters for Triplet network, 6.43 million parameters for Inception network, 30 million parameters for Capsule network [38]. While, we use in total 99*(N+1)+2 neurons to recognize fingerprints, where N is the total number of minutiae in template and test fingerprint and it is less than 100, and in each neuron there are one spiking rule and one synapse creating rule. There are 13 created synapses during the computation in each part, thus in total $13*8 = 104$ synapses are involved.

Compared with other neural networks, such as SPCNN and combined LVQ networks with accuracy 96.4 [39], PCASYS (PNN+ Pseudoridge) with accuracy 94.9% [40], and spiking neural networks based fingerprint identification with accuracy from 70% to 94.74% when applying on different datasets [41], our results are acceptable.

Also, we did the experiments on the other three databases and we calculate the EER of our algorithm. The results are shown as follows.

Compared with some existing algorithms [42]–[45], our EER is not perfect. The main reason is that we attach more importance on the building of SNP systems instead of accuracy-oriented systems. We tried to apply SNP systems to some real applications. More importantly the results show the feasibility of recognizing fingerprints by self-organized SN P systems. To our best knowledge, it is the first attempt of using SN P systems to do fingerprint recognition.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed double layers self-organized spiking neural P systems using anti-spikes are proposed, which expands layers of SN P Systems. Based on SN P systems we proposed, a model is established to describe minutiae. And matching algorithm, especially matching part, is improved through the minutiae model. Although the results are not good, the algorithm can beparallelized, which will boost the speed.

Also, we conclude some possible reasons for the results not being perfect.

(1) The pre-processing in this paper is rough. The key point of this paper is the model we proposed, so there are limits in the pre-processing method.

(2) Fake minutiae are also a reason for that. Bad minutiae are noise for the algorithm.

(3) The square we choose is 9×9, maybe it contains too little information matching needs.

(4) The template is not good enough because we just choose the first pictures of each group.

(5) The square to express minutiae may be limited, maybe an irregular shape will catch the key information of minutiae.

The main contribution of the paper is as follow:

(1) We design a double layers self-organized spiking neural P system with anti-spikes. The system can self-adaptively create and delete synapse between the neurons in different layers.

(2) We use the proposed SN P systems for fingerprint recognition by the spike trains emitted out of the output neurons. It is the first try to use SN P systems in fingerprint recognition.

In the future work, we will change the model size to test the best minutiae model. And learning mechanism will be introduced for the system to get a qualified template at first, thus matching will be more efficient. Besides, security related features are also very important to a fingerprint recognition system. We will test our system in different scenarios and improve our algorithm.

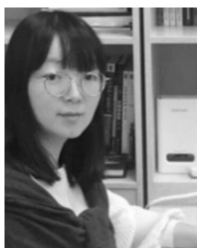
REFERENCES

[1] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, 1990.
[2] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*. Boston, MA, USA: PWS Publishing, 1996.
[3] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, Aug. 2009.
[4] C. Eurich and S. Wilke, "Multidimensional encoding strategy of spiking neurons," *Neural Comput.*, vol. 12, no. 7, pp. 1519–1529, Jul. 2000.
[5] R.-Z. Gao, Y.-J. Wang, J.-F. Lai, and H. Gao, "Neuro-adaptive fault-tolerant control of high speed trains under traction-braking failures using self-structuring neural networks," *Inf. Sci.*, vols. 367–368, pp. 449–462, Nov. 2016.
[6] A. Segev, D. Curtis, S. Jung, and S. Chae, "Invisible brain: Knowledge in research works and neuron activity," *PLoS ONE*, vol. 11, no. 7, 2016, Art. no. e0158590.
[7] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.
[8] M. Buibas, E. M. Izhikevich, B. Szatmary, and V. Polonichko, "Neural network learning and collaboration apparatus and methods," U.S. Patent 9 208 432, 2015.
[9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[11] M. Ionescu, G. Păun, and T. Yokomori, "Spiking neural P systems," *Fundam. Inf.*, vol. 71, no. 2, pp. 279–308, 2006.
[12] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.
[13] L. Pan and G. Paun, "Spiking neural P systems with anti-spikes," *Int. J. Comput., Commun. Control, IV*, vol. 3, pp. 273–282, 2009.
[14] V. P. Metta, K. Krithivasan, and D. Garg, "Computability of spiking neural P systems with anti-spikes," *Math. Natural Comput.*, vol. 8, no. 3, pp. 283–295, 2012.

[15] T. Song, L. Pan, K. Jiang, B. Song, and W. Chen, "Normal forms for some classes of sequential spiking neural P systems," *IEEE Trans. Nanobiosci.*, vol. 12, no. 3, pp. 255–264, Sep. 2013.
[16] T. Song and X. Wang, "Homogenous spiking neural P systems with inhibitory synapses," *Neural Process. Lett.*, vol. 42, no. 1, pp. 199–214, 2015.
[17] T. Song, L. Pan, and G. Păun, "Asynchronous spiking neural P systems with local synchronization," *Inf. Sci.*, vol. 219, no. 10, pp. 197–207, 2012.
[18] H. Peng, J. Yang, J. Wang, T. Wang, Z. Sun, X. Song, X. Luo, and X. Huang, "Spiking neural P systems with multiple channels," *Neural Netw.*, vol. 95, pp. 66–71, Nov. 2017.
[19] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Dynamic threshold neural P systems," *Knowl.-Based Syst.*, vol. 163, pp. 875–884, Jan. 2019.
[20] H. Peng and J. Wang, "Coupled neural P systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1672–1682, Jun. 2019.
[21] G. Păun, "Spiking neural p systems with astrocyte-like control," *J. Univ. Comput. Sci.*, vol. 13, no. 11, pp. 1707–1721, 2007.
[22] A. Binder, R. Freund, M. Oswald, and L. Vock, "Extended spiking neural P systems with excitatory and inhibitory astrocytes," *Proc. 5th Brainstorming Week Membrane Comput.*, 2007, pp. 63–72.
[23] L. F. Macias-Ramos and M. J. Pérez-Jiménez, "Spiking neural P systems with functional astrocytes," in *Membrane Computing* (Lecture Notes in Computer Science). Springer, 2012, pp. 228–242.
[24] X. Zeng, X. Zhang, and L. Pan, "Homogeneous spiking neural P systems," *Fundam. Inform.*, vol. 97, nos. 1–2, pp. 275–294, 2009.
[25] T. Song, X. Wang, Z. Zhang, and Z. Chen, "Homogenous spiking neural P systems with anti-spikes," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1833–1841, 2014.
[26] J. Wang, H. J. Hoogeboom, L. Pan, G. Păun, and M. J. Pérenz-Jiménez, "Spiking neural P systems with weights," *Neural Comput.*, vol. 22, no. 10, pp. 2615–2646, 2010.
[27] X. Zhang, B. Luo, X. Fang, and L. Pan, "Sequential spiking neural P systems with exhaustive use of rules," *BioSystems*, vol. 108, no. 1, pp. 52–62, 2012.
[28] X. Zhang, X. Zeng, B. Luo, and L. Pan, "On some classes of sequential spiking neural P systems," vol. 26, no. 5, pp. 974–997, 2014.
[29] M. Ionescu, G. Paun, and T. Yokomori, "Spiking neural p systems with an exhaustive use of rules," *Int. J. Unconventional Comput.*, vol. 3, no. 2, pp. 974–997, 2002.
[30] X. Zhang, B. Wang, and L. Pan, "Spiking neural P systems with a generalized use of rules," *Neural Comput.*, vol. 26, no. 12, pp. 2925–2943, 2014.
[31] T. Song and L. Pan, "Spiking neural P systems with rules on synapses working in maximum spikes consumption strategy," *IEEE Trans. Nanobiosci.*, vol. 14, no. 1, pp. 38–44, Jan. 2015.
[32] T. Song and L. Pan, "Spiking neural P systems with rules on synapses working in maximum spiking strategy," *IEEE Trans. Nanobiosci.*, vol. 14, no. 4, pp. 465–477, Jun. 2015.
[33] G. Zhang, H. Rong, F. Neri, and M. J. Pérez-Jiménez, "An optimization spiking neural P system for approximately solving combinatorial optimization problems," *Int. J. Neural Syst.*, vol. 24, no. 5, p. 8.
[34] T. Song, L. Pan, T. Wu, Z. Pan, M. L. D. Wong, and A. Rodríguez-Patón, "Spiking neural P systems with learning functions," *IEEE Trans. Nanobiosci.*, vol. 18, no. 2, pp. 176–190, Apr. 2019, doi: 10.1109/TNB.2019.2896981.
[35] X. Wang, T. Song, F. Gong, and P. Zheng, "On the computational power of spiking neural p systems with self-organization," *Sci. Rep.*, vol. 6, 2016, Art. no. 27624.
[36] T. Song, S. Pang, and S. Hao, "A parallel image skeletonizing method using spiking neural P systems with weights," *Neural Process. Lett.*, pp. 1–18, 2018.
[37] T. Lam and S. Nilsson, "Application of convolutional neural networks for fingerprint recognition," M.S. thesis, Lund Univ., Lund, Sweden, 2018.
[38] [Online]. Available: http://bias.csr.unibo.it/fvc2002/results/resultsAvg.asp
[39] L. Ji, Z. Yi, and X. Pu, "Fingerprint classification by SPCNN and combined LVQ networks," in *Proc. ICNC LNCS*, vol. 4221, 2006, pp. 395–398.
[40] G. T. Candela, "PCASYSA pattern-level classification automation system for fingerprints," NIST, Gaithersburg, MD, USA, Tech. Rep. NISTIR 5647, Aug. 1995.
[41] Y. Elmir, Z. Elberrichi, and R. Adjoudj, *Spike Neural Network based Fingerprint Identification*, 2009.
[42] W. J. Wong, A. B. J. Teoh, and Y. H. Kho, "Kernel PCA enabled bit-string representation for minutiae-based cancellable fingerprint template," *Pattern Recognit.*, vol. 51, pp. 197–208, 2016.
[43] Z. Jin, A. B. J. Teoh, T. S. Ong, and C. Tee, "Fingerprint template protection with minutiae-based bit-string for security and privacy preserving," *Expert Syst. Appl.*, vol. 39, no. 6, pp. 6157–6167, May 2012.
[44] M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1727–1737, Dec. 2012.
[45] W. J. Wong, A. B. J. Teoh, M. L. D. Wong, and Y. H. Kho, "Enhanced multi-line code for minutiae-based fingerprint template protection," *Pattern Recognit. Lett.*, vol. 34, no. 11, pp. 1221–1229, 2013.

TONGMAO MA received the graduation degree from the China University of Petroleum, Qingdao, China, in 2015. He is currently pursuing the Ph.D. degree in artificial intelligence with the Universidad Politécnica de Madrid, Madrid, Spain. His current research interests include artificial intelligence, membrane computing, and bioinformatics.

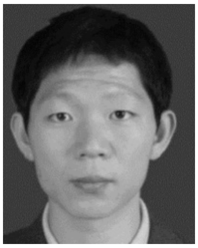
SHAOHUA HAO received the graduate degree from the China University of Petroleum, Qingdao, China. Her research interests include biocomputing and membrane computing.



XUN WANG received the graduate degree from the University of Tsukuba, Tsukuba, Japan. She is currently an Associate Professor with the China University of Petroleum, Qingdao, China. Her research interests include bioinformatics, artificial intelligence, biocomputing, big data mining, and membrane computing.

ALFONSO RODRÍGUEZ-PATÓN received the B.S. degree in electronic and computational from the University of Santiago de Compostela, Santiago de Compostela, Spain, in 1992, and the Ph.D. degree in computer science from the Polytechnic University of Madrid, Madrid, Spain, in 1999. He is currently an Associate Professor with the Faculty of Informatics, Polytechnic University of Madrid. His current research interests include interplay of computer science, biology, and engineering.

SHUDONG WANG received the graduation degree from the Huazhong University of Science and Technology, Wuhan, China, in 2004. She is currently a Professor with the China University of Petroleum, Qingdao, China. Her current research interests include DNA computing, large-scale graph data processing, and bioinformatics.



TAO SONG received the graduation degree from the Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently an Associate Professor with the China University of Petroleum, Qingdao, China, and a Visiting Fellow with the Polytechnical University of Madrid, Madrid, Spain. His current research interests include DNA computing, membrane computing, and bioinformatics.