Commun Nonlinear Sci Numer Simulat 17 (2012) 1637–1648

Contents lists available at SciVerse ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Application of new advanced CNN structure with adaptive thresholds to color edge detection

Shaojiang Deng [a], Yuan Tian [a,*], Xipeng Hu [b], Pengcheng Wei [c], Mingfu Qin [a]

[a] College of Computer science, Chongqing University, Chongqing 400044, China
[b] Chongqing Communication College, Chongqing 400035, China
[c] Department of Computer Science, Chongqing Education of College, Chongqing 400067, China

ARTICLE INFO

Article history:
Received 11 May 2011
Received in revised form 18 August 2011
Accepted 5 September 2011
Available online 14 September 2011

Keywords:
Cellular neural network (CNN)
Color edge detection
Detection threshold
Mahalanobis distance
Adaptive templates
Multilayer

ABSTRACT

Color edge detection is much more efficient than gray scale detection when edges exist at the boundary between regions of different colors with no change in intensity. This paper presents adaptive templates, which are capable of detecting various color and intensity changes in color image. To avoid conception of multilayer proposed in literatures, modification has been done to the CNN structure. This modified structure allows a matrix *C*, which carries the change information of pixels, to replace the control parts in the basic CNN equation. This modification is necessary because in multilayer structure, it faces the challenge of how to represent the intrinsic relationship among each primary layer. Additionally, in order to enhance the accuracy of edge detection, adaptive detection threshold is employed. The adaptive thresholds are considered to be alterable criteria in designing matrix *C*. The proposed synthetic system not only avoids the problem which is engendered by multi-layers but also exploits full information of pixels themselves. Experimental results prove that the proposed method is efficient.

1. Introduction

In computer vision, edges play important roles in many applications such as object recognition, tracking, image retrieval and so on. Compared with gray-scale pictures, color images generally include richer measurement information that can be successfully exploited to improve the performance of image-based instrumentation and/or extend its application range [1,2]. In 1987 Novak and Shafer found that 90% of the edges are about the same in gray value images and in color images [3]. However, there are still 10% of the edges left that may not be detected in intensity images, which may be important for a consecutive processing step such as edge-based image segmentation or edge-based matching. And what's more, when edges exist at the boundary between regions of different colors with no change in intensity, color edge detection schemes performs extremely better than gray scale approaches [7]. Although color-based edge detection includes more information than gray-based detection, it is much less well defined and has not received the same attention as gray-based ones [4,5]. One of the difficulties in edge detection in color image is the definition of what an edge is. Indeed, in gray-level images a scalar gray-level is assigned to a pixel of image, but in color images, a color vector which consists of several components is assigned to a pixel. Another difficulty is how to integrate the contrast information contained in various components into one meaningful result.

So far, monochromatic-based techniques of applying a gray-level algorithm to the single components of the image and then combining the obtained results are the most appealing edge detection methods. It should be observed that the processing of

* Corresponding author.
E-mail address: 63198038@qq.com (Y. Tian).

1007-5704/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cnsns.2011.09.007

three color channels is generally necessary to yield an accurate edge map. Indeed, edges in a color image can be produced by objects having the same (or similar) luminance but different chrominance information. Vector space approaches take into account this issue by modeling each image pixel as a three-dimensional vector in the assigned color coordinate system [6]. These methods generally adopt gradient-based algorithms that resort to an appropriate definition of a distance in the given color space. Another effective class of edge detection techniques is based on order statistics [8]. Basically these operators are characterized by linear combinations of rank-ordered pixel vectors. Operators with different performance and efficiency can be designed by appropriately choosing the set of coefficients in the linear combination. Difference vector (DV) methods are another powerful family of edge detectors: a gradient is typically computed in each of the four main directions and the maximum gradient vector is selected to detect edges [9].

Although these methods mentioned above are powerful in color detection, some important aspects are rarely addressed such as the complexity of the algorithm and their efficient hardware implementation. They are all very important in real time image processing. In recent years image processing based on cellular neural network (CNN) have achieved great success. The main advantage of using CNN in image processing is related to the large-scale non-linear analog circuits with real-time signal processing capability; especially it satisfies the high-speed parallel processing. It employs parallel computing paradigm defined in n-dimensional regular array of elements and came of the Hopfield neural networks and cellular automata as an effective combination of both characteristics [11,12]. Therefore it has two features: one is the continuous-time characteristics and we call it real-time signal processing capabilities. At the same time, the characteristics of its local interconnection make it applicable to VLSI realization. In image processing, a pixel's value is calculated based only on its neighbor pixels. Neural networks like Hopfield networks lacking this property of locality make them unsuitable for image processing applications [14]. As for image processing based on CNN, the key point is to find proper templates to complete different tasks. Different CNN templates can be used to carry out different phases of processing such as: direction detection, edge detection, black and white reverse [15], hole filter [25], shadow detection [26], logic operator, etc. The design of templates constitutes one of the crucial research problems in the field of CNN and it is a non-trivial problem.

There are mainly two ways to design the template: one way is first considering the parameters as a solution in optimization techniques and then taking advantage of training method in neural networks to train templates parameters in the given samples. There are some efficient training algorithms such as PSO [23], Tabu search [22], GA [21] and so on. In Li et al. [24] investigated another method: linear matrix inequality (LMI) for training templates. Indeed, using training method is an easy way to find desirable templates in samples; however, the templates have most possibility of over fitting the given samples and lack of generalization capability. The other way is directly according to the given task. For instance, in edge detection, we define a pixel as edge when there are more than three pixels satisfying such condition as the distance of central pixel and neighborhood greater than a threshold. However, because of complex background of color image, it is unrealistic to find a uniform threshold suitable for every component [17]. Moreover, this method faces another challenge of how to input the three-dimensional color image into the CNN dynamics, because one CNN layer can only process one dimensional data.

In this paper, as for template designing, a directly way is considered. Research in [10] has shown that the human vision have strong adaptive ability to the color and intensity change. It does not consider this aspect fully in cloning templates. Therefore, in order to get more accurate edge in color detections, adaptive templates are employed to detect edges in this paper. It is well known that one CNN layer can only process one dimension data. So in order to break this constraint, a new structure of CNN is proposed inspired by template designing mentioned in [25,26]. In [25,26], CNN templates for hole-filling and shadow detection are reported based on gray-scale scheme. It employed such a special way that considering the control parts in basic CNN equation to be a scalar. This special designing way can also make three-dimensional control parts in color detection be one dimensional data. So in this way, directly inputting three-dimensional image can be avoided. However, in color detection much more information than gray-scale should be taken into account. And an identical scalar cannot sufficient present information in color images which included in control parts. Observed that the change among neighboring pixels is the key aspect should be considered in edge detection. Therefore, to take full advantage of this special character, the control parts can be designed to be combining one. This one carries change information among pixels. On the other hand, to find proper thresholds discussed previously, adaptive thresholds are designed based on the human vision achievement. These adaptive templates can process color image with various color and intensity information because every template carries special color and intensity character of pixel's value. Compared with the method proposed in [27], which split green channel for vessel segment based on CNN (because of its special application to Medical), the synthetic system has more generality. Because templates designing in this synthetic system consider three primary channels fully. Finally in order to facilitate the simulation on a digital computer and make parameters more robust [19], the whole task is performed on discrete sequential CNN (DTCNN) architecture.

The structure of the present work is organized as follows. Section 2 introduces the mathematical model of CNN and DTCNN. And then, we descript the color space and introduce some of the researches based on the selected model using CNN and also determine the measurement of distance, and depict the process of the selection of adaptive threshold in Section 3. Section 4 is devoted to design the proper templates to fulfill color edge detection. Section 5 displays some results processed by the given method, followed by the conclusions in Section 6.

2. CNN mathematical model

In this Section, we consider CNNs composed by $M \times N$ cells and arranged on a regular grid. We represent a cell $c(i,j)$ using a variable i which denotes horizontal position and a variable j which denotes vertical position. Each cell only connects to its neighboring cells with the radius of r according to a template.

Cells neighborhood are shown as follows:

$$s_r(i,j) = \left\{ c(k,l) \middle| \max_{1\leqslant k\leqslant M, 1\leqslant l\leqslant N} \{\{|k-i|, |l-j|\}\} \leqslant r \right\}. \tag{1}$$

The state equation and the output equation of the cell are shown as follows:

$$\begin{aligned} \frac{dx_{i,j}(t)}{dt} &= -x_{ij} + \sum_{C_{i+k,j+l}\in S_r(i,j)} a_{k,l}(i,j,t) y_{i+k,j+l} + \sum_{C_{i+k,j+l}\in S_r(i,j)} b_{k,l}(i,j,t) u_{i+k,i+l} + z_{i,j}(i,j,t) \\ &= -x_{ij} + \sum_{k=-r}^{r} \sum_{l=r}^{r} a_{k,l} y_{i+k,j+l} + \sum_{k=-r}^{r} \sum_{l=r}^{r} b_{k,l} u_{i+k,j+l} + z_{i,j}; \end{aligned} \tag{2}$$

output equation:

$$y_{i,j} = f(x_{i,j}) = \frac{1}{2}(|x_{i,j}-1| - |x_{i,j}+1|), \quad i = 1,2,\ldots,M,\ j = 1,2,\ldots,N, \tag{3}$$

where $x_{i,j}$, $y_{i,j}$, $u_{i,j}$ and $z_{i,j}$ represent a state, an output, an input and a bias of a cell, r is a positive integer denoting the neighborhood radius of each cell, $a_{k,l}$, $b_{k,l}$ is feedback template A and control template B. It is easy to find from Eqs. (2) and (3) that the CNN has the features of spatial discreteness, nonlinearity and parallel processing capability.

The output function is showed in Fig. 1. The state of each cell is bounded for all time $t > 0$ and, after the transient has settled down, a class of cellular neural networks always approaches one of its stable equilibrium points. Furthermore, if the system satisfies that the center element of template A is greater than one, i.e., $A_{i,j} > 1$ then, the settled state values will converge to absolute values greater than one [13].

Lemma 1. $\frac{dx(t)}{dt} = \lim_{h\to 0} \frac{x(t+h)-x(t)}{h}$.

According to Lemma 1, Eq. (2) can be similarly rewritten as:

$$x_{i,j}(t+h) = h\left[-x_{ij}(t) + \sum_{C_{i+k,j+l}\in S_r(i,j)} a_{k,l}(i,j) y_{i+k,j+l}(t) + \sum_{C_{i+k,j+l}\in S_r(i,j)} b_{k,l}(i,j) u_{i+k,i+l} + z_{i,j}(i,j) \right] + x_{i,j}(t), \tag{2a}$$

$$y_{i,j}(t+h) = f(x_{i,j}(t+h)) = \frac{1}{2}(|x_{i,j}(t+h)-1| - |x_{i,j}(t+h)+1|), \tag{3a}$$

The models are called time discredited CNN (DTCNN). When the step-length is enough small, DTCNN can be similarly considered as continuous one. The DTCNN can easily be simulated on a digital computer, given that an integration algorithm is not required. Furthermore, from the point of view of hardware implementation, a DTCNN with a sufficiently high robustness parameter is quite insensitive to noise and variations in the parameters originated by the fabrication tolerances [20]. On the other hand, due to the fact that the processing of the DTCNN is synchronous, it is possible to establish a robust control over the propagation velocity [20].

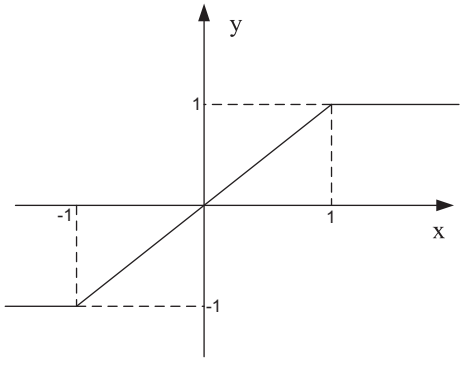

Fig. 1. The output function of CNN.

3. Color space and distance measurement

3.1. RGB space

To perform any kind of color image processing, a color model must be selected. With CNN there is no exception. The purpose of color model is to facilitate the specification of colors in some standard manner [14]. Basically, a color model is a specification of a three-dimensional coordinate system and a subspace within that system where each color is represented by a point. In this paper RGB model is selected which is employed by all computer hardware display. In this scheme a color is represented by the relative amounts of three primary colors that are required to produce the given color. Assume each pixel is represented by a vector $\vec{C}$.

$$\vec{C} = [c_r, c_g, c_b]^T = [c_r, c_g, c_b]^T = [R, G, B]^T. \tag{4}$$

For example, in an $M \times N$ image, there are $M \times N$ vectors like $\vec{C}$ as follows:

$$\vec{C} = \begin{bmatrix} c_r(x,y) \\ c_g(x,y) \\ c_b(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix},$$

where $x = 1,2,\ldots,M$ and $y = 1,2,\ldots,N$.

3.2. Distance measurement

As for RGB space, there are two approaches about how to process color images based on CNN. One is that allocate a CNN layer to each primary color component, carry out the processing independently, and then form the triplet to see the results. It is called *sequential color processing* mode. For instance, a color image processing model is proposed in [14]. It takes the advantage of RGB model that each primary color can be represented by a CNN layer, e.g., red, green and blue layers (L_r,L_g,L_b). To be able to work with multiple layers, it expands the basic CNN equation to a matrix equation as follows:

$$\frac{dx_{i,j}(t)}{dt} = -x_{i,j}(t) + \sum_{c(k,l)\in N_r(i,j)} \overline{A}(i,j;k,l)y_{k,l}(t) + \sum_{C(k,l)\in N_r(i,j)} \overline{B}(i,j;k,l)u_{k,l} + I, \tag{2c}$$

$$y_{i,j}(t) = \frac{1}{2}(|x_{i,j}(t)+1| - |x_{i,j}(t)-1|); \tag{3c}$$

where $\overline{A}$ and $\overline{B}$ are block triangular matrices and I, x and y are vectors as follows:

$$\overline{A} = \begin{bmatrix} A_r & 0 & 0 \\ A_{rg} & A_g & 0 \\ A_{rb} & A_{gb} & A_b \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} B_r & 0 & 0 \\ B_{gb} & B_g & 0 \\ B_{rb} & B_{gb} & B_b \end{bmatrix},$$

$$x = \begin{bmatrix} x_{rij} \\ x_{gij} \\ x_{bi,j} \end{bmatrix}, \quad y = \begin{bmatrix} y_{rij} \\ y_{gij} \\ y_{bij} \end{bmatrix}, \quad u = \begin{bmatrix} u_{rij} \\ u_{gij} \\ u_{bij} \end{bmatrix} \text{ and } I = \begin{bmatrix} I_{rij} \\ I_{gij} \\ I_{bij} \end{bmatrix}.$$

Although they use A_{rg} and A_{gb}, etc. to reflect the relationship between primary colors, every state variable is independent of each other. This model faces the challenge of how to find these proper multi-templates. Although this model was illustrated in 1996, so far as we know, there are few methods for designing the multi-template in literature.

The counterpart is a *concurrent color processing* mode, this mode results from applications in which it is not desired to split a pixel's color into its three basic components. In this case, it would be based on the Euclidean distance, or any other norm, of the three RGB values [14]. Works in this paper are based on this mode. In edge detection, change among pixels is the key aspect need to be considered, rather than the pixel values. So take advantage of this character fully, B templates can be avoided directly designing. The control parts can be integrated into one part C, which carries change information. In this case, challenge of finding proper multi-templates is avoided (detail discuss can be found in Section 5). In RGB space, the change information among pixels can be transferred into distance information among vectors. So this method resorts to an appropriate definition of vectors. The most common and simper measurement is Euclidean distance. Let $\vec{a}(a_r, a_g, a_b)$ and $\vec{b}(b_r, b_g b_b)$ be two points in space. The Euclidean distance $D(a,b)$ of $\vec{a}$ and $\vec{b}$ is described as follows:

$$D(\vec{a},\vec{b}) = \|a-b\| = [(\vec{a}-\vec{b})^T(\vec{a}-\vec{b})]^{\frac{1}{2}} = [(a_r-b_r)^2 + (a_g-b_g)^2 + (a_b-a_b)^2]^{\frac{1}{2}}. \tag{5}$$

Note that in Euclidean distance, each counterpart of vector plays the same role. However in color display the three primary parts are not the same weight. Hence a proper way should be found to weigh the different weight of each primary. A variant of Euclidean distance called *Mahalanobis* distance has the ability of weighing; it can be described as follows:

Table 1
Three primary colors vision functions (normalized).

Light color	Wavelength/nm	Version function
Blue	440	0.114
Green	500	0.587
Red	660	0.299

$$D(\vec{a},\vec{b}) = [(\vec{a}-\vec{b})^T C(\vec{a}-\vec{b})]^{\frac{1}{2}}, \tag{6}$$

where C is a diagonal matrix, the elements of the main diagonal is the weights of each component.

On the other hand, Three-Primary-Vision functions (PVF) display the weight of the three primary colors [16]. It can be found in Table 1. According to PVF the normalized value are sorted to be the main diagonal value. Let other elements be zero, matrix $C = diag\{0.299, 0.587, 0.144\}$.

4. Adaptive thresholds

The retina is composed of two major classes of photoreceptor cells known as rods and cones because of the shapes of their outer segments. Rods are extremely sensitive to light and provide achromatic vision at low illumination levels. Cones are sensitive than the rods but provide color vision at high levels. The luminance change cannot be detected by both cells until it reaches certain degree the detection threshold relative to background luminance. We can describe this relationship known as Weber law.

$$s = k \lg I + k_0, \tag{7}$$

where s is subjective luminance value, I is objective one, k and k_0 is constant. More researches on the relationship have been done in [10]. They have plotted the detection threshold against the corresponding background luminance called threshold-versus-intensity (TVI) function as in Fig. 2 [10].

There is a modeled luminance threshold detection function of Rods according to Fig. 2; it is described as follows [18]:

$$\lg \Delta I = \begin{cases} \lg I - 2.8 & -6 \leqslant \lg I \leqslant -4, \\ \lg I - 2.9 & -4 < \lg I < -3, \\ -0.449 + \lg I \exp(0.0665 \lg I) & -3 \leqslant \lg I < -1.5, \\ \lg(0.4467 I) & -1.5 \leqslant \lg I \leqslant 2.33. \end{cases} \tag{8}$$

It is also pointed that the luminance detection threshold of gray and green-cones are basically the same [18]. So Eq. (7) can be used to represent the gray-cones luminance threshold detection. And in color image processing, detection threshold in color appearance should be sufficient explained. So they use golden partition law to extend the arguments in Eq. (7) to the range of [0–255]. This extending-rule mainly based on Weber-law and some particular points to get the value k in Eq. (7) (more detail can be got in [18]). They worked out the color threshold detection threshold of green:

$$\Delta g = \begin{cases} 12734 \exp(-0.1494 g) & 0 \leqslant g \leqslant 37, \\ 5107.5 \exp(-0.1015 g) & 37 < g < 60, \\ 120470 \exp[(0.07 g - 7.55) \exp(0.0026 g) - 0.089 g] & 60 \leqslant g < 97, \\ 8.0031 & 97 \leqslant g \leqslant 255 \end{cases} \tag{9}$$

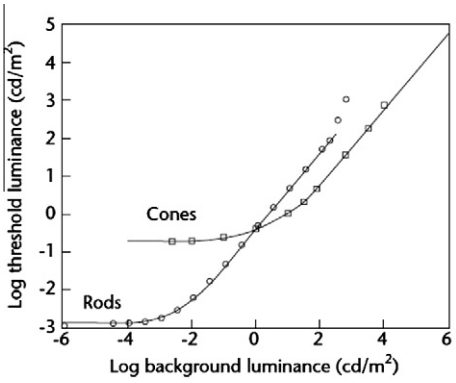

Fig. 2. TVI function curve.

and they also worked out the detection thresholds of red and blue which are described in Eqs. (10) and (11). (The detailed process was proposed in [18].)

$$\Delta r = \begin{cases} 12734\exp(-0.1494r) & 0 \leqslant g \leqslant 37, \\ 5397\exp(-0.1015r) & 37 < g < 60, \\ 127300\exp[(0.07r - 7.55)\exp(0.0026r) - 0.089r] & 60 \leqslant g < 97, \\ 8.4569 & 97 \leqslant g \leqslant 255, \end{cases} \tag{10}$$

$$\Delta b = \begin{cases} 12734\exp(-0.1494b) & 0 \leqslant g \leqslant 37, \\ 9104.6\exp(-0.1015b) & 37 < g < 60, \\ 214750\exp[(0.07b - 7.55)\exp(0.0026b) - 0.089b] & 60 \leqslant g < 97, \\ 14.2663 & 97 \leqslant g \leqslant 255. \end{cases} \tag{11}$$

Eqs. (9)–(11) and Fig. 2 all indicate that human detection threshold has strong adaptation. So in edge detection it is not suitable to use an identical threshold for all pixels. To agree with the *Mahalanobis distance* value mentioned in Section 3, the three detection thresholds should be synthesized into one value. Some transformations are taken in the following way according to structure of *Mahalanobis distance*. First square the thresholds of R, G, B, and then add weight. Let $t_{i,j}$ be the threshold of $p_{i,j}$, it can be described as follows:

$$t_{i,j} = w_r\Delta r_{i,j}^2 + w_g\Delta g_{i,j}^2 + w_b\Delta b_{i,j}^2, \tag{12}$$

where $\Delta r_{i,j}$, $\Delta g_{i,j}$ and $\Delta b_{i,j}$ get from Eqs. (9)–(11). Let $w_r = 0.299$, $w_g = 0.587$, $w_b = 0.144$ according to PVF mentioned in Section 3.

5. Adaptive CNN templates design

From the previous argument, adaptive thresholds of pixels and measurement of distance among pixels have been obtained. In this Section, designing algorithm to color edge detection is to be addressed. Analyze the basic CNN equation in detail, two parts of equation are unchanged during interactions, they are: $b_{k,l}u_{i+k,j+l}$ and $z_{i,j}$. So they can be considered as constants during the interactions. Templates can be designed in such a way as: calculate control parts to be a constant before interaction, and then input the constant into CNN interaction instead of directly inputting control parts (three dimensional vectors) during interactions. Then, the structure of multilayer is avoided. Let $C_{k,l}$, which carries change information, replace the part $b_{k,l}u_{i+k,j+l}$, a three dimensional vector in the basic CNN equation. In order to get proper templates to color edge detection, edge detection tasks are defined as follows:

(1) Global task
Given: a static color image P;
Input: $U(t) = P$;
Boundary condition: cycling boundary condition;
Initial state: $y(t) \Rightarrow y(\infty)$.
(2) Local rules

$u_{i,j}(0) \to y_{i,j}(\infty)$;

One: arbitrary → black (−1) if there are at least three nearest neighbors satisfying $|\Delta u| > t$;
Two: arbitrary → white (1) if there are at most two nearest neighbors satisfying $|\Delta u| < t$;

where Δu is the *Mahalanobis*-space-distance among pixels in RGB space and it is described as follows:
$\Delta u = \left[(u_{i,j} - u_{i+k,j+l})^T D(u_{i,j} - u_{i+k,j+l})\right]^{\frac{1}{2}}$, where $(u_{i,j} - u_{i+k,j+l}) = (u(r)_{i,j} - u(r)_{i+k,j+l}u(g)_{i,j} - u(g)_{i+k,j+l}, \quad u(b)_{i,j} - u(b)_{i+k,j+l})$; $D = diag\{0.299, 0.587, 0.144\}$; t is the threshold according to Eq. (12). Use $C_{k,l}$ to replace $b_{k,l}u_{i+k,j+l}$ in the basic CNN equation which can be described as follows:

$$\frac{dx_{i,j}(t)}{dt} = -x_{ij} + \sum_{k=-r}^{r}\sum_{l=r}^{r} a_{k,l}y_{i+k,j+l} + \sum_{k=-r}^{r}\sum_{l=r}^{r} c_{k,l} + z_{i,j}. \tag{13}$$

Let radius $r = 1$ and a kind of uncoupled CNN can be defined. The template has the form as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} c_{-1,-1} & c_{-1,0} & c_{-1,1} \\ c_{0,-1} & c_{0,0} & c_{0,1} \\ c_{1,-1} & c_{1,0} & c_{1,1} \end{bmatrix}, \quad Z = z, \tag{14}$$

where $c_{k,l} = \begin{cases} s & \Delta u \geqslant t \\ -1 & \Delta u < t \end{cases}$.

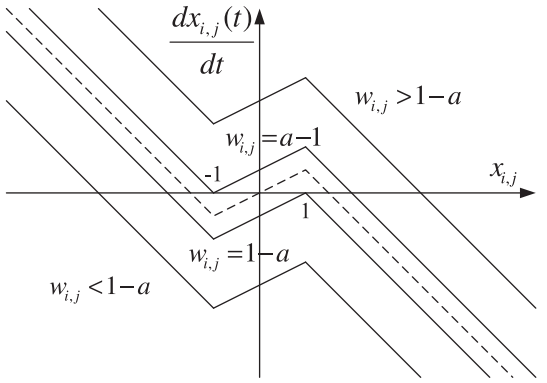

Fig. 3. The dynamic function of CNN.

Lemma 2. *If the center element of template A is greater than one i.e. $A_{i,j} > 1$, the settled state values will converge to absolute values greater than one [13].*

Theorem 1. *If s, z satisfying the following conditions (15), (16), then templates defined in (14) can perform in robustness way according to the edge detection task.*

$$3s + z > 5, \tag{15}$$

$$2s + z < 6. \tag{16}$$

Robustness analysis: from Eqs. (2) and (3), it follows:

$$\frac{dx_{i,j}(t)}{dt} = -x_{ij} + ay_{i,j} + w_{i,j}, \tag{17}$$

$$= \begin{cases} -x_{i,j} + a + w_{i,j} & x_{i,j} \geqslant 1, \\ (a-1)x_{i,j} + w_{i,j} & -1 < x_{i,j} < 1, \\ -x_{i,j} - a + w_{i,j} & x_{i,j} \leqslant -1, \end{cases} \tag{18}$$

where $w_{i,j} = \sum_{k=-1}^{1}\sum_{l=-1}^{1} c_{k,l} + z_{i,j}$.

According to Lemma 2 and local rule, we assume $a > 1$. Fig. 3 is the state dynamic function.

Analyze Eqs. (2) and (18), when $x_{i,j} \geqslant 1$, the output $y_{i,j}$ is steadily equal to 1, $y_{i,j}(\infty) = 1$. When $-1 < x_{i,j} < 1$, there are three cases: (1) when $w_{i,j} > a - 1$, $x_{i,j}(\infty)$ will converge to great than 1, $y_{i,j}(\infty) = 1$ with no initial condition requirement. (2) when $w_{i,j} < 1 - a$, $x_{i,j}(\infty)$ will converge to less than -1, $y_{i,j}(\infty) = -1$, with on initial condition requirement; (3) when $1 - a < w_{i,j} < a - 1$, if $x_{i,j}(0) > -\frac{a-1}{w_{i,j}}$, $y_{i,j}(\infty) = 1$, if $x_{i,j}(0) < -\frac{a-1}{w_{i,j}}$ $y_{i,j}(\infty) = -1$. The analyzed results can be described as follows:

$$y_{i,j} = \begin{cases} 1 & w_{i,j} > a - 1 & x_{ij}(0) \in (-\infty, \infty), \\ 1 & 1 - a \leqslant w_{ij} \leqslant a - 1 & x_{ij}(0) > -\frac{a-1}{w_{i,j}}, \\ -1 & 1 - a \leqslant w_{ij} \leqslant a - 1 & x_{ij}(0) < -\frac{a-1}{w_{i,j}}, \\ -1 & w_{i,j} < 1 - a & x_{ij}(0) \in (-\infty, \infty). \end{cases} \tag{19}$$

Remark: Under the condition that $1 - a \leqslant w_{i,j} \leqslant a - 1$, when $x_{i,j}(0) = -\frac{a-1}{w_{i,j}}$ or $x_{i,j}(0) = 0$, the output $x_{i,j}(\infty)$ will not converge to the equilibrium point, so these conditions should be avoided.

In order to simplify the analysis, assume $a = 2$ and $x_{i,j}(0) = 0$. In this case, Eq. (17) can be written as follows:

$$y(\infty) = \begin{cases} 1 & w_{i,j} > 0, \\ -1 & w_{i,j} < 0. \end{cases} \tag{20}$$

$s\Delta u > t$, p_s be the number of pixels $\Delta u > t$, then $p_d = 8 - p_s$. According to Eq. (18) and first satisfy the local rule one: $p_s > 3$, $w_{i,j} = p_s s + (8 - p_s) \times (-1) + z \geqslant 3s - 5 + z > 0$; second, in order to satisfy local rule two: $p_s < 3$, $w_{i,j} = p_s s + p_d \times (-1) + z \leqslant 2s - 6 + z < 0 \leqslant 2s - 6 + z < 0$. The two local rules can be formulized as follows: $3s + z > 5$; $2s + z < 6$. Analysis has been completed.

Theorem 2. *In the uncoupled CNN system, convergence is considered to be reached under the condition that all state values $|x_{i,j}(t)| > 1$.*

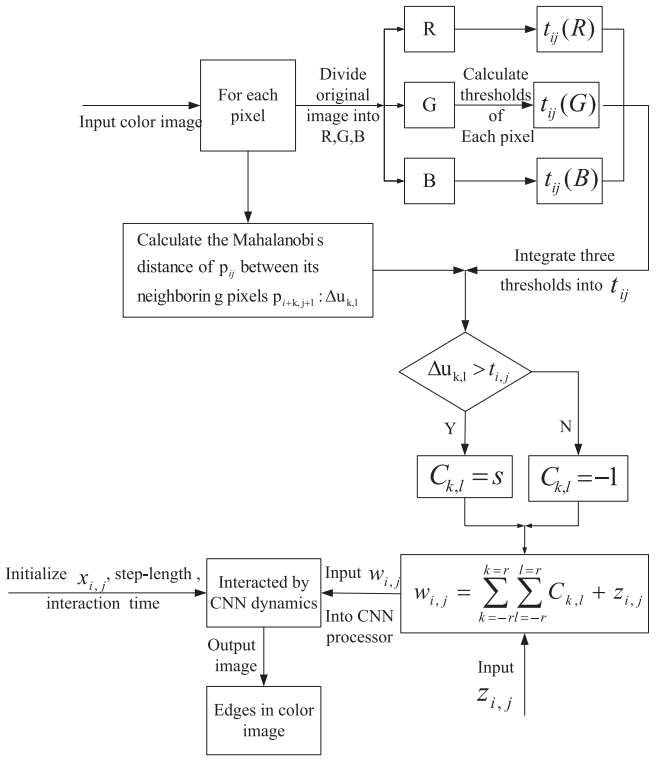

Fig. 4. The algorithm flow chart.

Proof. In this system, look closely at Eq. (17); each cell current output is independent of each other after $w_{i,j}$ has determined. So when each cell converges respectively, system is consequently converged. Considering one cell reaches convergence in this system under the condition that $\frac{dx_{i,j}(t)}{dt} = 0$. When $x_{i,j}(t) > 1, y_{i,j}(t) = 1, \frac{dx_{i,j}(t)}{dt} = -x_{i,j} + ay_{i,j} + w_{i,j} = -x_{i,j} + a + w_{i,j}$, as long as $x_{i,j}(t) = a + w_{i,j}$, according to Eq. (20) when $y_{i,j}(t) = 1$, $w_{i,j} > 0$, there exists $w_{i,j}$ such that $\frac{dx_{i,j}(t)}{dt} = 0$; when $x_{i,j}(t) < -1$, it can be similarly proved.

In the DTCNN structure, we can also use Eqs. (19) and (20) to get C template according to the relationship between CNN and DTCNN. The whole algorithm flow chart is shown in Fig. 4.

6. Simulation results

In this Section, several results of experiments are described. The simulations of edge detection are all under the platform Matlab7.0. Choose $s = 2$, $z = 1$, $x_{i,j}(0) = 0$ based on the argument previously. According to Theorem 2, we consider the convergence as reached when all state values $|x_{i,j}(t)| > 1$.

Firstly, in order to prove the efficiency of the proposed method in edge detection, two other classic methods (*Sobel*,*Prewitt*) are used as reference. Fig. 5(a) presents the original color image Lena, a standard color test image. Fig. 5(b) is extracted edges by performing this proposed method, Fig. 5(c) and (d) present extracted edges obtained by *Sobel* and *Prewitt*. It can be

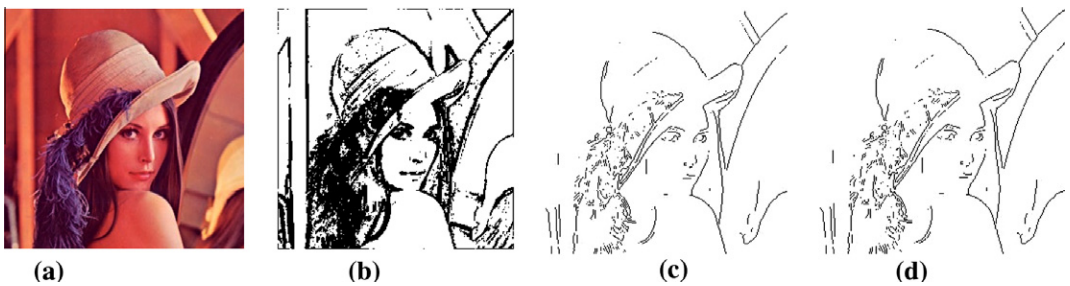

Fig. 5. Results of edge detection: (a) the original color image "Lena", (b) detected result of the proposed method, (c) detected result of *Sobel*, (d) detected result of *Prewitt*.

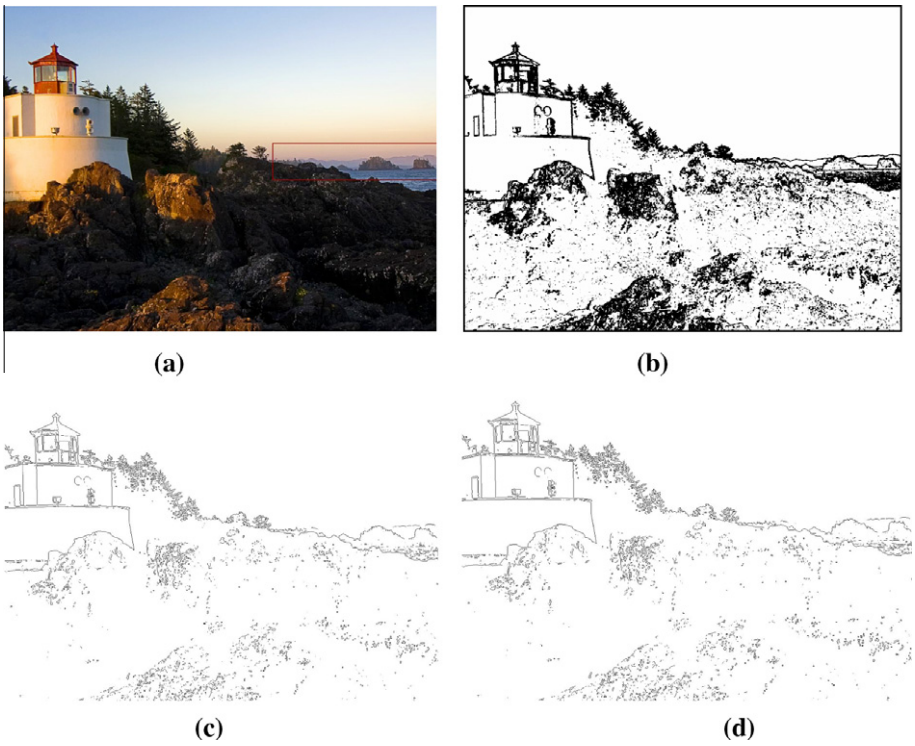

Fig. 6. Results detected by different algorithms: (a) the original image "Lighthouse", (b) proposed method, (c) *Sobel*, (d) *Prewitt*.

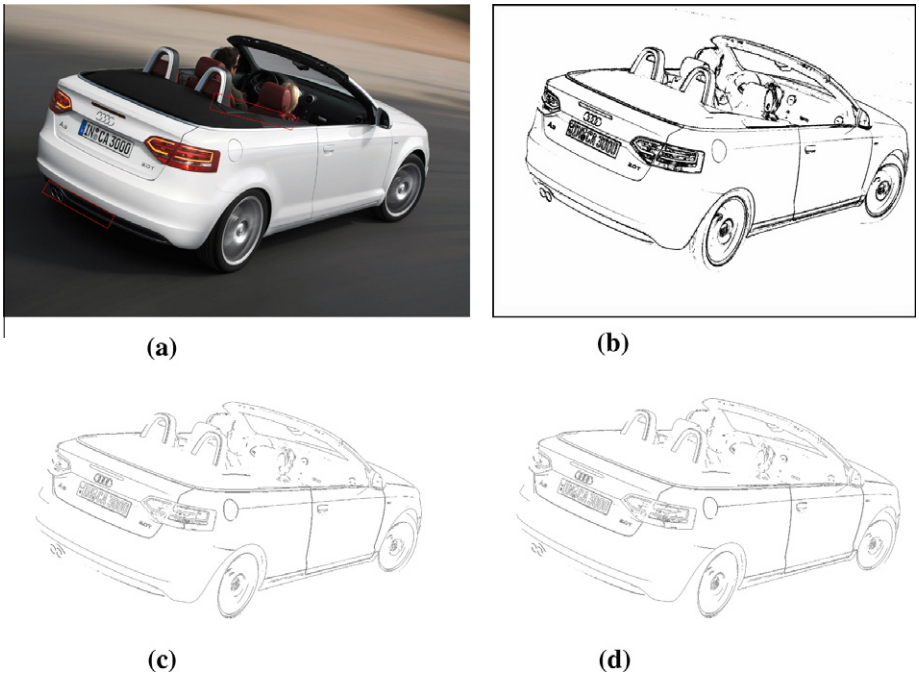

Fig. 7. Results detected by proposed method: (a) the original color image "Audi", (b) proposed method, (c) *Sobel*, (d) *Prewitt*.

found that this proposed color detector can provide more accurate results; some obvious edges of background in the image are missed when performed by *Sobel* and *Prewitt*. Additionally, the extracted edges performed by this proposed method include more detail change which can be specially seen in the face edges.

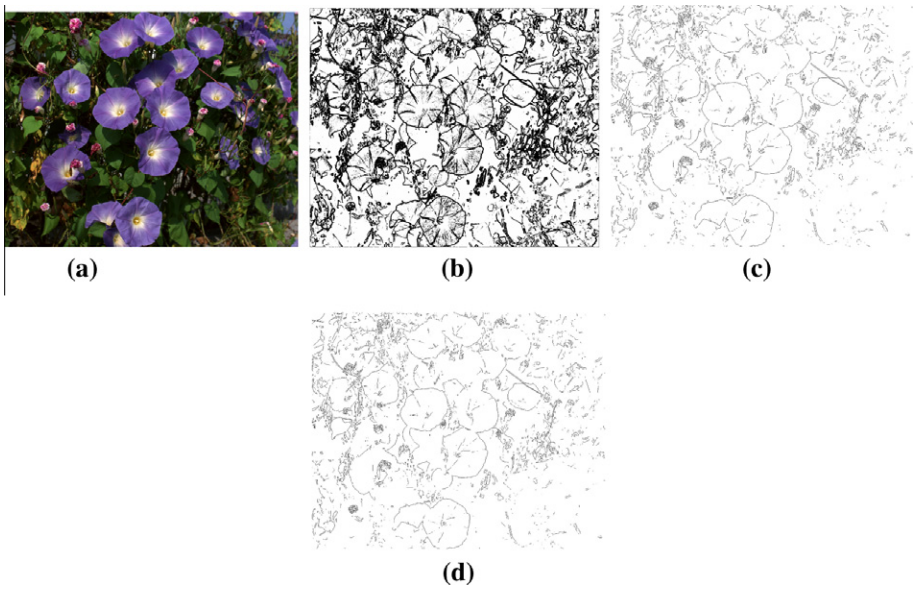

Fig. 8. Results detected by different algorithms: (a) the original image "morning glory", (b) proposed method, (c) *Sobel*, (d) *Prewitt*.

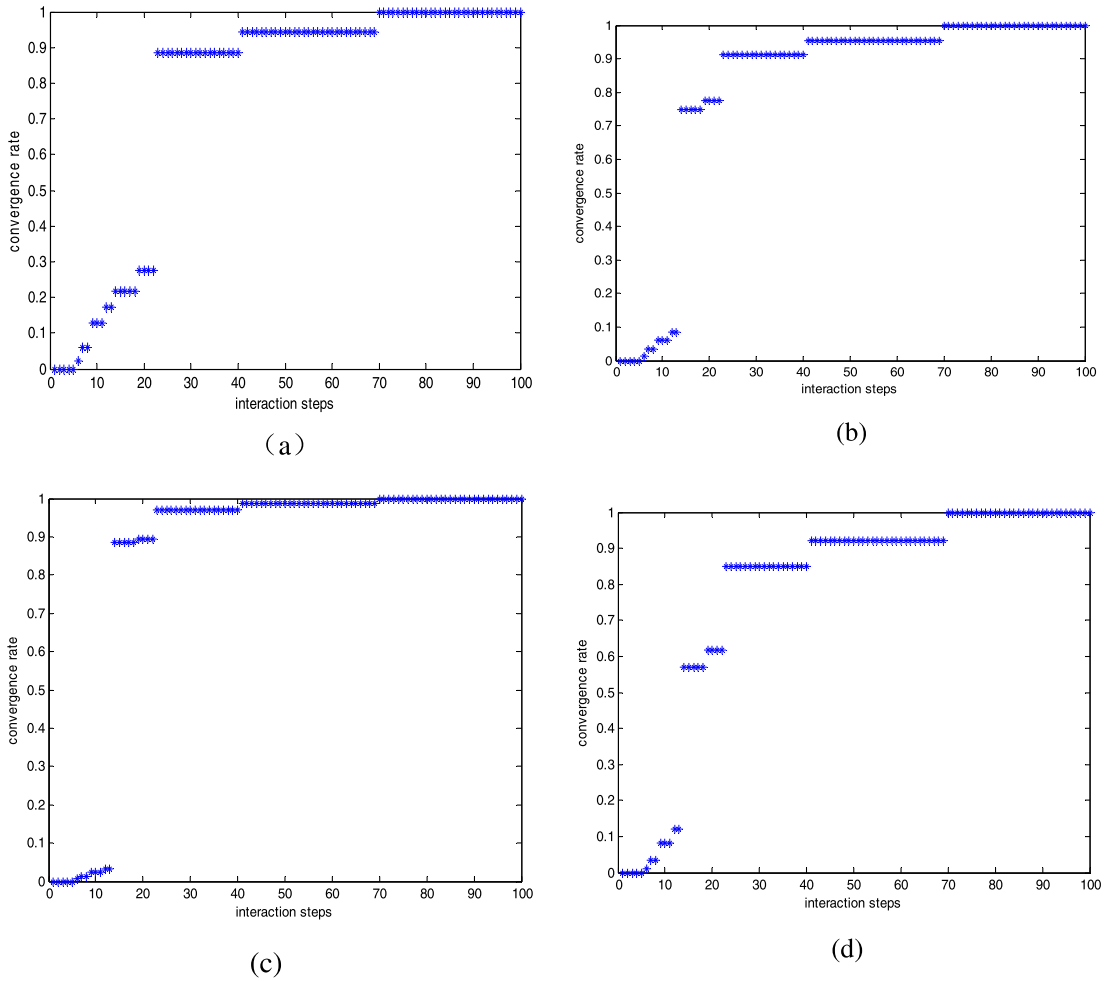

Fig. 9. Convergence function for each processed image: (a) Lena, (b) Lighthouse, (c) Audi, (d) Morning glory.

Second, in order to obtain a quantitative evaluation of the performance of the proposed method, three other real images are to be tested: the 1024 × 768 Lighthouse image (Fig. 6(a)) with various color and luminance change, the 1920 × 1200 Audi image (Fig. 7(a)) which represent single simple object, the 1280 × 1024 Morning Glory (Fig. 8(a)), a common color image, also with full color and luminance change. Fig. 6(b)–(d) are the extracted edges by performing this proposed method, *Sobel* and *Prewitt* respectively. Compare these extracted edges; this proposed method obtains more continuous and accurate edge line. Obvious edges missed by performing *Sobel* and *Prewitt* are highlighted in red box in original image. Fig. 7(b)–(d) and Fig. 8(b)–(d) are the extracted edges from Audi and Moring Glory respectively. As the same of Lighthouse, some missed edges by performing the two classic detectors are highlighted in Audi image in red box. As for edges of Morning Glory, we can easily find the proposed method provides more unambiguous edge than two classic detectors and also can detect detail change of the flower. Experimental results have shown the strong adaptability and steady detecting capability of this synthetic system to color detection. One can find, under the same conditions, the proposed method can provide more potential edges as compared with these well-known techniques.

As we know, the main advantage of CNNs is their low processing time. So in order to show the time efficiency of this synthetic system, testing on velocity of convergence is to be done. Convergence rate can be defined as follows:

$$Convergence\ rate = \frac{n}{N}, \tag{21}$$

where N is the number of total pixels, n is the number of pixels that obtain equilibrium points. Fig. 9(a)–(d) present the convergence process of Lena, Lighthouse, Audi and Morning Glory respectively. x-Axes are the interaction steps and y-Axes are convergence rate defined in Eq. (21). It can be seen in Fig. 9 that testing images convergence rapidly during the first 20 steps, and after 70 interaction steps, all of these images reach convergence. Fig. 9 proves time efficiency of this synthetic system.

7. Comments and conclusions

A new synthetic system for color edge detection based on CNN has been presented. The proposed system avoids the constraint of multilayer structure in color image detection based on CNN. It has avoided directly inputting multi-dimension data by using change information of pixels to be control parts. Therefore, it transforms multi-dimensional data into one dimension. Additionally, thresholds have been determined according to human vision system. Therefore, it makes the templates more suitable for pixels. Experimental results have shown that the proposed method performs better than classic methods such as *Sobel* and *Prewitt* in color image detection. The designing method of templates in this paper also gives us inspiration: in some special tasks using CNN, modification to CNN structure can be done according to specific task. In this way, we need not directly designing proper A, B templates.

Acknowledgement

The work described in this paper is supported by the Natural Science Foundation of China (No. 61173178), by Natural Science Foundation Project of CQ CSTC (NO. 2009BB2227), the Foundation of Chongqing Education Committee (Nos. KJ091501, KJ091502, KJ101501, KJ101502).

References

[1] Pitas I. Digital image processing algorithms and applications. New York: Wiley; 2000.
[2] Mitra SK, Sicuranza G. Nonlinear image processing. New York: Academic; 2000.
[3] Novak CL, Shafer SA. Color edge detection. In: Proc. DARPA image understanding workshop, LosAngeles, CA, USA; 1987. p. 35–37.
[4] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986;8(6):679–98.
[5] Marr D, Hildreth E. Theory of edge detection. Proc Roy Soc London B 1980;207:187–217.
[6] Machuca R, Phillips K. Applications of vector fields to image processing. IEEE Trans Pattern Anal Mach Intell 1983;PAMI-5(3):316–29.
[7] Evans Adrian N, Liu Xin U. A morphological gradient approach to color edge detection. IEEE Trans Image Process 2006;15(6).
[8] Trahanias PE, Venetsanopoulos AN. Vector order statistics operators as color edge detectors. IEEE Trans Syst Man Cybern 1996;26:135–43.
[9] Zhu S-Y, Plataniotis KN, Venetsanopoulos AN. Comprehensive analysis of edge detection in color image processing. Opt Eng 1999;38(4):612–25.
[10] Ferwerda JA. Element of early vision for computer graphics [J]. IEEE Comput Graph Appl 2001;21(5):22–33.
[11] Chua LO, Yang L. Cellular neural networks: theory. IEEE Trans Circuits Syst 1988;35(10):1257–72.
[12] Roska T, Chua LO. CNN universal machine: an anagogic array computer. IEEE Trans Circuits Syst 1993;40(3):163–73.
[13] Chua L0, Yang L. Cellular neural networks: theory. IEEE Trans Circuits Syst 1988;CAS-35:1257–72.
[14] Lee Chi-Chien, Gyvez Jose Pineda de. Color image processing in a cellular neural-network environment. IEEE Trans neural networks 1996;7(5).
[15] Gonzalez Rafael C, Woods Richard E. Digital image processing [M]. Beijing China: Electronic Industry Press; 2003.
[16] Chuang Zhang, J-nan Chi. a novel color edge detector derived from CNN mode. Comput Eng Appl 2008;44(21):17–9.
[17] Shuhai Chen, Luxiang Fu, Practical Digital Image Processing [M]. Beijing: Science Press China.
[18] guang-yu Zhang, Mei Xie. A new algorithm of color images edge detection. J UEST China 2005;34(2):164–7.
[19] Harrer H, Nossek. Discrete-time cellular neural networks. Int J Circuit Theor Appl 1992;20:453–67.
[20] Harrer H, Nossek JA, Stelzl. An analog implementation of discrete-time cellular neural networks. IEEE Trans Neural Networks 1993;3:466–76.
[21] Kozek T, Roska T, Chua LO. Genetic algorithm for CNN template learning. IEEE Trans Circuit Systems-I 1993;40(6):392–402.
[22] Li Chunguang, Xu Hongbing, Liao Xiaofeng. Tabu search for CNN template learning. Neurocomputing 2003;51:475–9.
[23] Fornarelli Girolamo, Giaquinto Antonio. Adaptive particle swarm optimization for CNN associative memories design. Neurocomputing 2009;72:3851–62.
[24] Li Huaqing, Liao Xiaofeng, Li Chuandong, Huang Hongyu, Li Chaojie. Edge Detection of noisy images based on cellular neural networks. Commun Nonlinear Sci Numer Simul 2011;16(7):3746–59.

[25] Matsumoto T, Chua LO, Furukawa R. CNN cloning template: hole-filler. IEEE Trans Circuits Syst 1990;37(5):635–8.
[26] Matsumoto T, Chua LO, Furukawa R. CNN cloning template: shadow detector. IEEE Trans Circuits Syst 1990;37(8):1070–3.
[27] Perfetti Renzo, Ricci Elisa, Casali Daniele, Costantini Giovanni. Cellular neural networks with virtual template expansion for retinal vessel segmentation. IEE Trans Circuits System-II: Express Briefs 2007;54(2).