Image Processing Using Periodic Pattern Formation in Cellular Neural Networks

Taisuke Nishio * Yoshifumi Nishio *

Abstract — **In this paper, an image processing algorithm using periodic pattern formation in cellular neural networks is proposed. Applying this algorithm to a certain image, area having spatial uniformity is removed from the original image and only parts without spatial uniformity remain. Simulation results show that when this algorithm is applied to an image containing characters (texts), only characters are extracted.**

1 INTRODUCTION

Cellular neural networks (CNNs) were introduced by Chua and Yang in 1988 [1]. The idea of the CNN was inspired from the architecture of the cellular automata and the neural networks. Unlike the conventional neural networks, the CNN has local connectivity property. Since the structure of the CNN resembles the structure of animals' retina, the CNN can be used for various image processing applications including character extractions [2][3]. Also, the CNN is known to be able to generate periodic patterns [4][5].

It is amazing to imagine how the eye of animals performs image processing. Although the importance of computer vision has been recognized widely, it is still difficult to develop a computer vision system with the performance of animals' eye. In computer vision, at first, a feature of images is extracted, and next, various image processing and detailed analyses are performed on its basis. However, it is difficult to develop a universal feature extraction method for various kinds of images in real world. Hence, it is important to devise more general feature extraction method, which does not deeply depend on a type of images.

In this study, as a new approach for image processing, we propose a method using periodic pattern formation in the CNN. At first, we investigate various CNN templates producing periodic patterns. Next, we develop an image processing algorithm using one of the templates, which makes some spots only in the area having spatial uniformity. Simulation results show that when this algorithm is applied to images containing characters (texts), only characters are extracted. Note that the purpose of this study is not to develop an algorithm extracting characters in images, but to find out some possible image-processing applications using periodic pattern formation of the CNN.

*Department of Electrical and Electronic Engineering, Tokushima University, 2-1 Minami-Josanjima, Tokushima 770-8506, Japan. Email: {taisuke, nishio}@ee.tokushima-u.ac.jp,
Tel.: +81-88-656-7470, Fax: +81-88-656-7471.

2 CELLULAR NEURAL NETWORKS

The fundamental circuit unit of CNN is called a **cell**. CNN consists of an array of cells. Each cell is connected only to its neighboring cells according to a cloning template. A cloning template is the same for all cells except for boundary cells. CNNs have features of time continuity, spatial discreteness, nonlinearity and parallel processing capability. CNNs can perform rapid and various image processing.

2.1 CNN Model

A simple CNN model is shown as follows.

State Equation:

$$\begin{aligned}\frac{dv_{xij}(t)}{dt} &= -v_{xij}(t) \\ &+ \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{(i,j;k,l)}v_{ykl}(t) \\ &+ \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{(i,j;k,l)}v_{ukl}(t) \\ &+ I \qquad (1)\end{aligned}$$

Output Equation:

$$v_{yij}(t) = \frac{1}{2}(|v_{xij}(t)+1| - |v_{xij}(t)-1|) \qquad (2)$$

v_x, v_y and v_u represent a state, an output and an input of cell, respectively. $\boldsymbol{A}$ is the feedback template and $\boldsymbol{B}$ is the control template, these and bias I are collectively called **cloning template**.

2.2 The r-neighborhood of a cell

Consider a two-dimensional array composed of $\boldsymbol{M} \times \boldsymbol{N}$ cells arranged in $\boldsymbol{M}$ rows and $\boldsymbol{N}$ columns. The generic cell placed on the $\boldsymbol{i}$th row and the $\boldsymbol{j}$th column is denoted by $\boldsymbol{C(i,j)}$, then, the **r-neighborhood** of a cell $\boldsymbol{C(i,j)}$ is defined as follows.

$\boldsymbol{N_r(i,j)}$

$$= \{C(k,l)|max\{|k-i|, |l-j|\} \leq r, \quad 1 \leq k \leq M; \quad 1 \leq l \leq N\} \quad (3)$$

where r is a positive integer number.

Figures 1 and 2 show the $r = 1$ (3×3) neighborhood and the $r = 2$ (5×5) neighborhood, respectively.

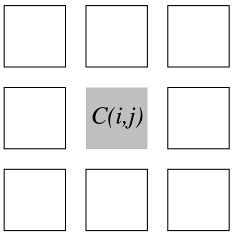


Figure 1: the $r = 1$ neighborhood.

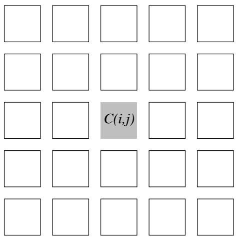


Figure 2: the $r = 2$ neighborhood.

3 PERIODIC PATTERN FORMATION IN CELLULAR NEURAL NETWORKS

In this section, the developed cloning templates for periodic pattern formation and the simulation results are shown.

3.1 CHECKERBOARD Template

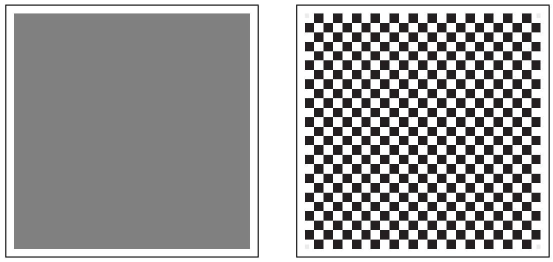
This template forms the black and white alternate pattern with the cycle of 2 cells. Although the alternate pattern with the cycle of 1 cell can be formed easily, the checkerboard patterns with the cycle of more than 1 cell has not been reported. We found that it is possible to form the checkerboard patterns with the cycle of more than 1 by increasing the size of the template.

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} -0.1 & -1 & -3 & -1 & -0.1 \\ -1 & 0.3 & -2 & 0.3 & -1 \\ -3 & -2 & -4 & -2 & -3 \\ -1 & 0.3 & -2 & 0.3 & -1 \\ -0.1 & -1 & -3 & -1 & -0.1 \end{bmatrix} \\ \boldsymbol{B} &= 0 \\ \boldsymbol{I} &= 0 \end{aligned} \quad (4)$$

3.2 RIPPLE Template

This template forms a pattern like a ripple.

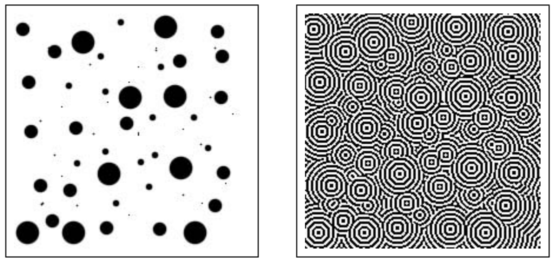
$$\boldsymbol{A} = \begin{bmatrix} 0 & -0.2 & -0.2 & -0.2 & 0 \\ -0.2 & 0 & 0.2 & 0 & -0.2 \\ -0.2 & 0.2 & 1.5 & 0.2 & -0.2 \\ -0.2 & 0 & 0.2 & 0 & -0.2 \\ 0 & -0.2 & -0.2 & -0.2 & 0 \end{bmatrix}$$



(a) (b)

Figure 3: CHECKERBOARD Template $(50 \times 50$ $cells)$. (a) Initial state. (b) Output.

$$\begin{aligned} \boldsymbol{B} &= 0 \\ \boldsymbol{I} &= 0 \end{aligned} \quad (5)$$



(a) (b)

Figure 4: RIPPLE Template $(200 \times 200$ $cells)$. (a) Initial state. (b) Output.

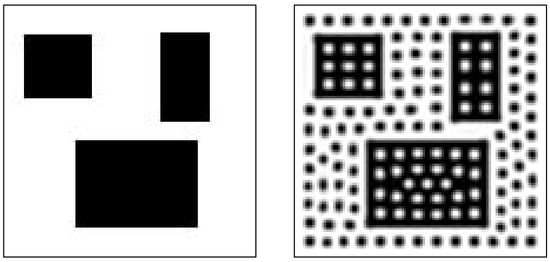
3.3 SPOT Template

This template makes white spots in a black area having spatial uniformity and black spots in a white area having spatial uniformity. It is interesting that the spots are made only in areas wider than a certain size.

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} -0.25 & -1 & -1.5 & -1 & -0.25 \\ -1 & 2.5 & 7 & 2.5 & -1 \\ -1.5 & 7 & -23.25 & 7 & -1.5 \\ -1 & 2.5 & 7 & 2.5 & -1 \\ -0.25 & -1 & -1.5 & -1 & -0.25 \end{bmatrix} \\ \boldsymbol{B} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{I} &= 0 \end{aligned} \quad (6)$$

4 IMAGE PROCESSING

In this section, an image processing algorithm using the pattern formation with the SPOT template is developed.



(a) (b)

Figure 5: SPOT Template **(100 × 100 *cells*)**. (a) Initial state and input. (b) Output.

The proposing algorithm is shown in Fig. 6. The **SPOT template** is used in this algorithm to form periodic pattern. The parenthetic numbers in the chart correspond to the parenthetic numbers in Fig 7. The other templates used in this algorithm are shown in Appendix.

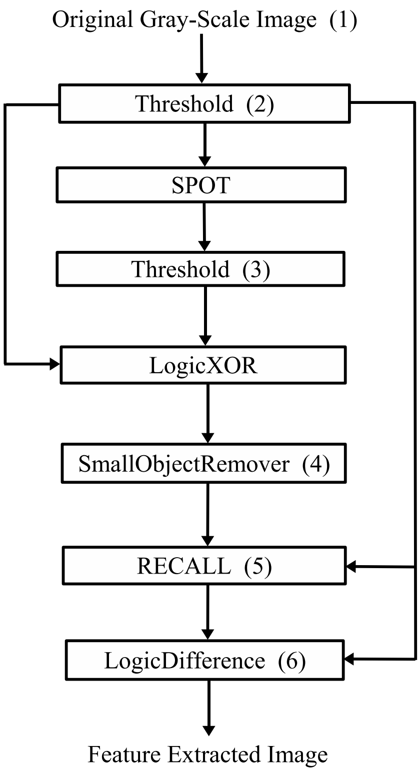


Figure 6: Image processing algorithm.

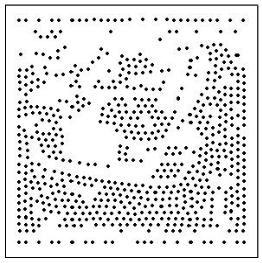
The simulation results of this algorithm for an image of a car and an image of road sign are shown below.

Figure 7 shows how the image is processed in the proposed algorithm. Figure 8 shows the image after noise removal from Fig. 7(6). This result shows that only the number on the number plate can be extracted by the algorithm.

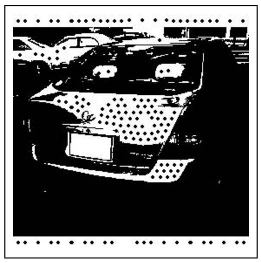
(1) original image.

(2) binarized image.

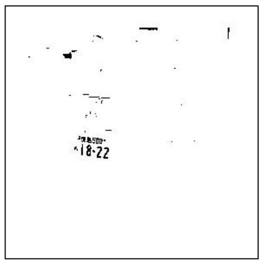
(3) pattern formation.



(4) pattern extraction.



(5) recalling the area with pattern.



(6) removing (5) from (2).

Figure 7: Simulation results for an image of a car.

Figure 8: Character extraction from a car image.

Figure 9 shows the original image of a road sign in Japan. Figure 10 shows the image after applying our algorithm to the image in Fig. 9. This result shows that only the important information on the road sign can be extracted by the algorithm.

Figure 9: Original gray-scale image of a road sign.

Figure 10: Character extraction from a road sign image.

5 CONCLUSIONS

In this study, we proposed an image processing algorithm using periodic pattern formation in CNNs. By computer simulations, we confirmed that this algorithm could extract only important characters from an image of a car and a road sign.

References

[1] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory," *IEEE Trans. Circuits Syst.*, vol.35, no.10, pp.1257-1272, Oct. 1988.

[2] T.Kozek, K.R.Crounse, T.Roska, and L.O.Chua, "Smart Image Scanning Algorithms for the CNN Universal Machine," *International Symposium on Nonlinear Theory and its Applications*, vol.2, pp.707-712, 1995.

[3] Jun Kishida, Csaba Rekeczky, Yoshifumi Nishio and Akio Ushida, "Feature Extraction of Postage Stamps Using an Iterative Approach of CNN" *IEICE Transactions on Fundamentals*, vol. E79-A, no. 10, pp. 1741-1746, Oct. 1996.

[4] K.R. Crounse and L.O. Chua, "Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial," *IEEE Trans. Circuits Syst.*, vol.42, no.10, pp.583-601, Oct. 1995.

[5] K.R. Crounse, L.O. Chua, P.Thiran and G.Setti, "Characterization and Dynamics of Pattern Formation in Cellular Neural Networks," *International Journal of Bifurcation and Chaos*, vol.6, no.9, pp.1703-1724, 1996.

A Other Templates Used in the Algorithm

A.1 Threshold Template

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = \boldsymbol{t} \tag{7}$$

A.2 Small Object Remover Template

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = \mathbf{0} \tag{8}$$

A.3 Recall Template

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 4 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = \mathbf{3} \tag{9}$$

A.4 Logic Difference Template

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = -\mathbf{1} \tag{10}$$