Output Characteristics of Cellular Neural Networks Using Mixture Template

Takashi Inoue, Masaru Nakano, Yoshifumi Nishio

***Abstract*— In this research, we propose cellular neural networks using mixture template as an example of space-varying cellular neural networks. As the first step of the investigation of such complex nonlinear circuit networks, we propose two mixing methods of the templates and investigate the output characteristics of the simple image processing with a binary image and a grayscale image by computer simulations.**

I. Introduction

CELLULAR Neural Networks (CNN) [1] were introduced by Chua and Yang in 1988. 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 application [2]-[4] including character extractions [5][6]. Further, the CNN can be utilized to produce some kinds of pattern generation [7][8].

Wiring weights of the cells are established by parameters called the template. The performance of the CNN is decided by the template. If the templates of all the cells in the CNN are identical, the system is called space-invariant, while if the templates of all the cells in the CNN are not identical, the system is called space-varying. Typically, space-invariant system is mainly used in the studies of the CNN, because the implementation cost is cheap. However, on the flip side, the performance of the system is considered to be limited.

In this research, we propose the CNN with mixture template as an example of space-varying CNN. By mixing two different templates, we expect that the proposed CNN possesses the both effects of the two templates. It may be useful to reduce the number of the successive precessing for some kinds of image processing tasks. Further, we are interested in the complex behaviors of such a new class of nonlinear circuit networks. They can be utilized for future engineering applications. As the first step of the investigation of this concept, in this research, we propose two mixing methods of the templates and investigate the output characteristics of the simple image precessing with a binary image and a grayscale image.

In the Sec. 2, we review the basic of the standard CNN. In the Sec. 3, we show how to mix the templates and show some computer simulated results. In the Sec. 4, noise removal performance of the proposed mixture templates is evaluated. In the Sec. 5, simulation results for a grayscale image are shown. In the Sec. 6, simulation results for a texture image are shown. Section 7 concludes the article.

II. Cellular neural networks[1]

In this section, we describe the basic structure of the CNN. The CNN has M by N processing unit circuits called cells as Fig. 1. Cells are arranged in a reticular pattern to M line N row. We represent a cell $C(i,j)$ using a variable i which denotes vertical position and a variable j which denotes horizontal position. The cell contains linear and nonlinear circuit elements. The CNN is an array of cells. Each cell is connected to its neighboring cells according to a template. Usually, the template is the same for all cells except for boundary cells. The CNN has the features of time continuity, spatial discreteness, nonlinearity and parallel processing capability.

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

State equation:

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

Output equation:

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

where v_x, v_y and v_u represent a state, an output and an input of cell, respectively. In the equation (1), A is the feedback template and B is the control template. These and bias I are collectively called general template.

The $r-neighborhood$ of the cell $C(i,j)$ in the CNN is defined by

$$\begin{aligned}Nr(i,j) = \{C(k,l)\,|\, &\max\{\,|k-i|\,,\,|l-j|\,\} \le r,\\ &1 \le k \le M; \quad 1 \le l \le N\} \qquad (3)\end{aligned}$$

where r is a positive integer number. In our study, we fix the value of r as 1.

III. Mixture template

Usually, it is desirable that all of the templates at all cells are identical, namely, space-invariant. However, we are interested in what kind of phenomena can be observed from space-varying CNN, because such systems are new class of nonlinear circuit networks and they may have a possibility for novel engineering applications.

Department of Electrical and Electronic Engineering, Tokushima University, 2-1 Minami-Josanjima, Tokushima 770-8506, Japan (email: {takashi, nakano, nishio}@ee.tokushima-u.ac.jp).

978-1-4244-1821-3/08/$25.00©2008 IEEE

In this research, we propose the CNN with two kinds of mixture templates, matrix mixing and element mixing, as an example of space-varying CNN,

A. Matrix mixing

In the first method of the mixing, we prepare two different CNN templates and choose one of them for each cell at random.

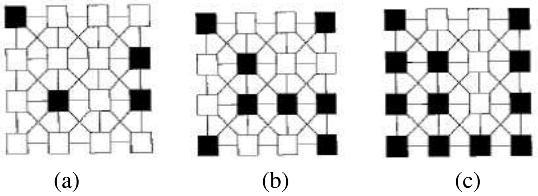


Fig. 1. Examples of matrix mixing. Black and white cells correspond to two different CNN templates. (a) $white : black = 75 : 25$. (b) $white : black = 50 : 50$. (c) $white : black = 25 : 75$.

Figure 1 shows structure examples of the matrix mixing. In this figure, black cells and white cells have different templates. Though we assume that the mixing is completely random in this article, the mixing method can be modified to some kinds of regular mixing method like alternate patterns. Also the ratio of black cells and white cells is decided optionally.

We carry out computer simulations of the matrix mixing using "Small Object Remover" template and "Patch Maker" template.

$Small\ Object\ Remover$:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = -1. \quad (4)$$

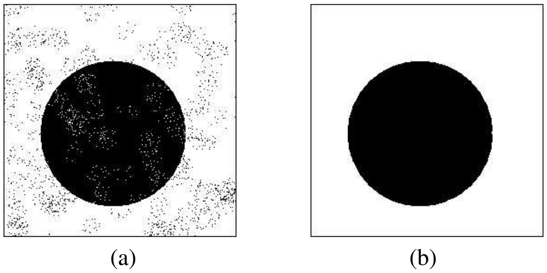


Fig. 2. The performance of Small Object Remover. (a) Input image. (b) Output image.

Figure 2 shows how the Small Object Remover template works. By using this template, large objects remain and small objects are removed from input binary images.

$Patch\ Maker$:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 4.5. \quad (5)$$

Figure 3 shows how the Patch Maker template works. By using this template, black objects are expanded without limit.

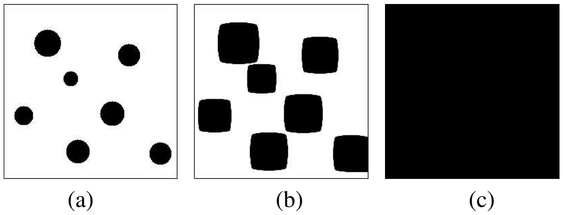


Fig. 3. The performance of Patch Maker. (a) Input image. (b) Transitional image. (c) Output image.

Figure 4 shows the simulation results using only Small Object Remover template for an input image with relatively large spot noises. We can see that the Small Object Remover cannot remove such noises at all.

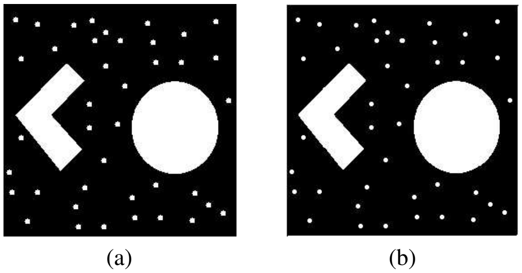


Fig. 4. Simulation results using only Small Object Remover template. (a) Input image. (b) Output image.

Figure 5 shows the simulation results for the same input image using the mixture template with the matrix mixing. From the output images, we can say that the CNN with the mixture templates possesses the both effects of the two templates, namely black regions are expanded by the effect of the Patch Maker template and the shrinking spot noises are removed by the effect of the Small Noise Remover template. Although the effect of the noise removal is enhanced by increasing the ratio of the Patch Maker template to the Small Object Remover template, it makes the shape of the large objects broken and the output becomes completely black for Figs. 5(d) and (e).

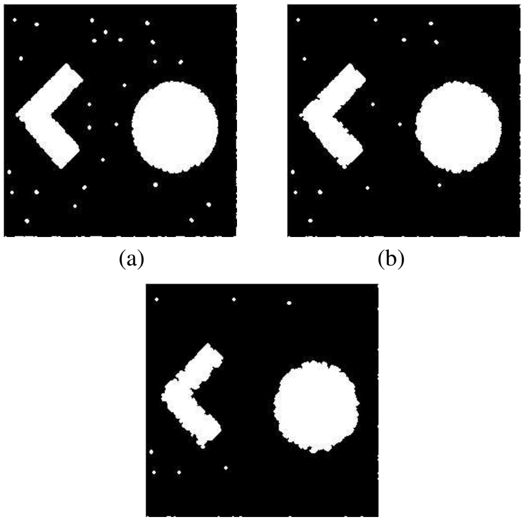
B. Element mixing

In the second method of the mixing, we prepare two different CNN templates as the matrix mixing, but choose the element of them for each element of the matrices and the threshold I at random as shown in Figs. 6 and 7.

Some examples of the mixture templates obtained from the Small Object Remover template and the Patch Maker template using the element mixing are as follows.

Example 1:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 4.5. \quad (6)$$



(a) (b)

(c)

Fig. 5. Simulation results for the same input image in Fig. 4(a) using mixture template with matrix mixing. (a) small object remover : patch maker= 80:20. (b) small object remover : patch maker= 70:30. (c) small object remover : patch maker= 60:40.

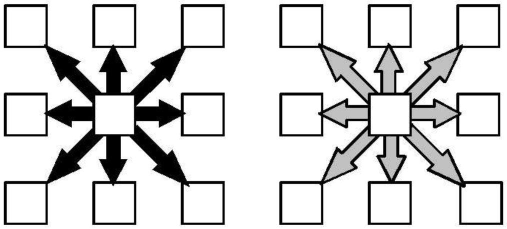


Fig. 6. Wiring weights of two templates.

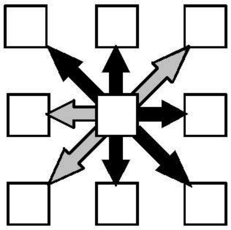


Fig. 7. Wiring weights of element mixing.

Example 2:

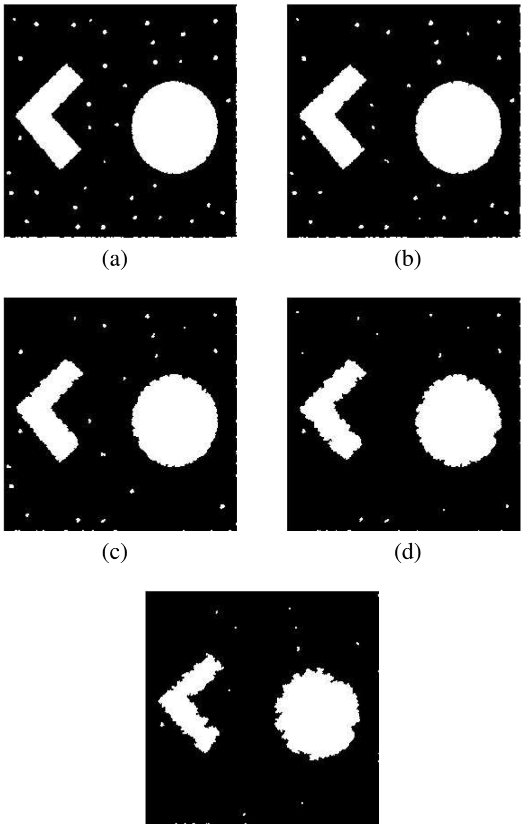
$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = -1. \quad (7)$$

Example 3:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 4.5. \quad (8)$$

The above templates are obtained by mixing the elements with even ratio.

Figure 8 shows the simulation results for the same input image as in the previous subsection using the mixture template with the element mixing of different mixing ratios. From the output images, we can say that this element mixing method has similar effect to the previous matrix mixing method.



(a) (b)

(c) (d)

(e)

Fig. 8. Simulation results for the same input image in Fig. 4(a) using mixture template with element mixing. (a) small object remover : patch maker= 80:20. (b) small object remover : patch maker= 70:30. (c) small object remover : patch maker= 60:40. (d) small object remover : patch maker= 50:50. (e) small object remover : patch maker= 40:60.

IV. Noise removal performance

In order to compare the two mixing methods quantitatively, we examine the noise removal performance investigating the denoising level of the small objects and the distortion level of the large objects of the input image. The input image have 40 noises. The Large objects of input image consist of 11514 pixels.

The denoising level of the small objects are evaluated by the number of denoising, namely, the number of the vanishing spot noises. While the distortion level of the large objects are evaluated by the number of the cells in the

TABLE I

NOISE REMOVAL PERFORMANCE OF MIXTURE TEMPLATES.

	Matrix mixing		Element mixing	
template (4) : (5) = 80:20	number of denoising	15	number of denoising	4
	distortion level	783	distortion level	663
template (4) : (5) = 70:30	number of denoising	26	number of denoising	10
	distortion level	1015	distortion level	808
template (4) : (5) = 60:40	number of denoising	33	number of denoising	20
	distortion level	1438	distortion level	1033
template (4) : (5) = 50:50	number of denoising	40	number of denoising	26
	distortion level	11514	distortion level	1394
template (4) : (5) = 40:60	number of denoising	40	number of denoising	30
	distortion level	11514	distortion level	2116

inputted large objects whose colors change from white to black, namely the difference in pixels between the original object images and the output object images.

The results are summarized in Table 1 for various mixing ratios. The both mixing methods remove the noise as the ratio of the Patch Maker increases. However, the distortion level becomes large. We can see that the effect of the matrix mixing seems to be larger than that of the element mixing.

V. SIMULATION RESULTS FOR GRAYSCALE IMAGE

In this section, we apply CNN with mixture template of the matrix mixing to a grayscale image in Fig. 9. In this simulation. we use the Smoothing Operation template and the Diffusion template.

$Smoothing\ Operation$:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 0. \tag{9}$$

$Diffusion$:

$$A = \begin{bmatrix} 0.1 & 0.15 & 0.1 \\ 0.15 & 0 & 0.15 \\ 0.1 & 0.15 & 0.1 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$I = 0. \tag{10}$$

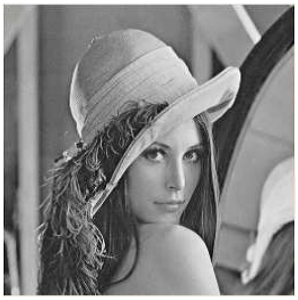
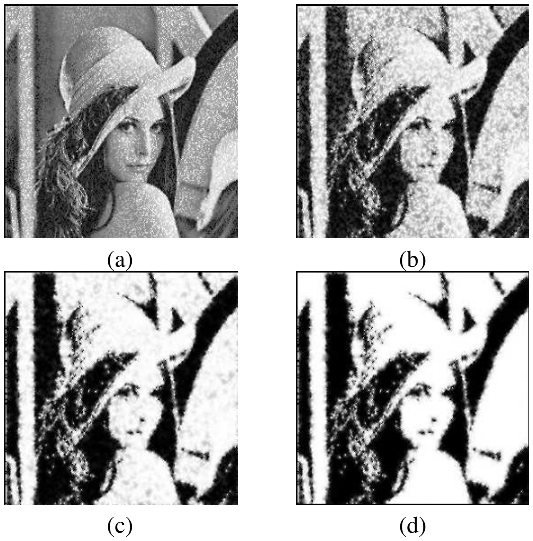


Fig. 9. Grayscale input image.

Figure 10 shows the transient variation of the output images when the mixing ratio of (9) and (10) is fixed as 20 : 80. Figure 11 shows the converged output images with changing the mixing ratio of (9) and (10). As a result, if we can choose suitable combination of multiple templates, then the image processing with both characteristics, namely smoothing and diffusion, can be realized,



(a) (b) (c) (d)

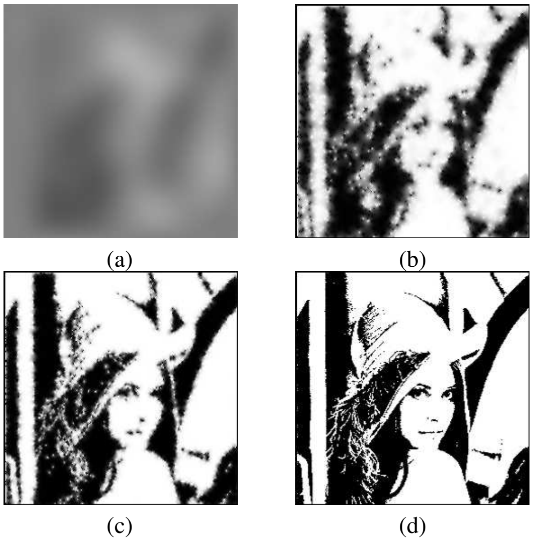
Fig. 10. Variation of output when mixing ratio of (9) and (10) is 20 : 80. (a) t =1. (b) t =5. (c) t =15. (d) t =20.

VI. SIMULATION REUSLTS FOR TEXTURE IMAGE

In this section, We apply CNN with mixture template of the matrix mixing to a texture image.

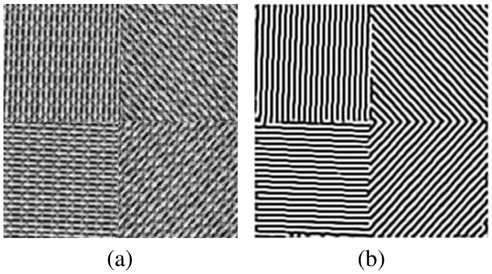
First, texture image is processed by SPOT template[9]. Figure 12 shown simulation results using SPOT template. By using this template, periodic pattern is emphasized.

Second, template that detects four directions is mixed. (ex. “Vertical Line Detect Template” and “Horizon Line Detect Template” is mixed.) Firure 13 shown how the template that detects four directins works. Figure 14 shown simulation results using mixture template with matrix mixing. The mixing rate is 50:50. As a results, when two types templates that detect one direction are mixed, two directions can be detected.



(a) (b) (c) (d)

Fig. 11. Converged output images with changing mixing ratio of (9) and (10). (a) (9) : (10) = 0 : 100. (b) 5 : 95. (c) 20 : 80. (d) 100 : 0.



(a) (b)

Fig. 12. (a) Texture image. (b) Simulation results using SPOT template.

SPOT Template:

$$\begin{aligned}
A &= \begin{bmatrix} -0.25 & -1 & -1.5 & -1 & -0.25 \\ -1 & 2.5 & 7 & 2.5 & -1 \\ -1.5 & 7 & -23.25 & 7 & -1.5 \\ -1 & 2.5 & 7 & 2.5 & -1 \\ -0.25 & -1 & -1.5 & -1 & -0.25 \end{bmatrix} \\
B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
I &= 0. \qquad (11)
\end{aligned}$$

Vertical Line Detect Template:

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} -1 & 0.5 & 1 & 0.5 & -1 \\ -1 & 1 & 1 & 1 & -1 \\ -1 & -1 & 5 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 \\ -1 & 0.5 & 1 & 0.5 & -1 \end{bmatrix} \\
I &= -13. \qquad (12)
\end{aligned}$$

Horizon Line Detect Template:

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ 0.5 & 1 & -1 & 1 & 0.5 \\ 1 & 1 & 5 & 1 & 1 \\ 0.5 & 1 & -1 & 1 & 0.5 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \\
I &= -13. \qquad (13)
\end{aligned}$$

Diagonally Right Up Line Detect Template:

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} -1 & -1 & -1 & 0.5 & 1 \\ -1 & -1 & 1 & 1 & 0.5 \\ -1 & 1 & 5 & 1 & -1 \\ 0.5 & 1 & 1 & -1 & -1 \\ 1 & 0.5 & -1 & -1 & -1 \end{bmatrix} \\
I &= -13. \qquad (14)
\end{aligned}$$

Diagonally Left Up Line Detect Template:

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 1 & 0.5 & -1 & -1 & -1 \\ 0.5 & 1 & 1 & -1 & -1 \\ -1 & 1 & 5 & 1 & -1 \\ -1 & -1 & 1 & 1 & 0.5 \\ -1 & -1 & -1 & 0.5 & 1 \end{bmatrix} \\
I &= -13. \qquad (15)
\end{aligned}$$

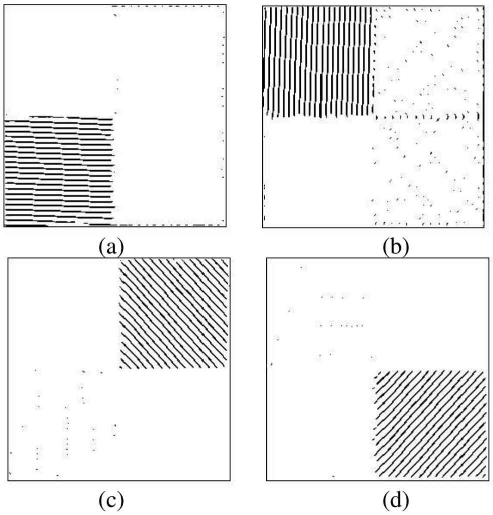


Fig. 13. Simulation results for the input image in Fig. 12(b). (a) Using Vertical Line Detect. (b) Using Horizon Line detect. (c) Using Diagonally Right Up Line Detect. (d) Using Diagonally Left Up Line Detect.

VII. Conclusions

In this research, we have proposed the CNN using mixture template with two kinds of mixing methods, the matrix mixing and the element mixing. By computer simulations of simple image processing tasks for a binary image and a grayscale image and texture image, we investigated the basic output characteristics of the proposed systems. As we expected, we could confirm the effects of the both templates by mixing them to one mixture template. At the moment, we do not say that the proposed method exhibited a superior performance than the original CNN at the moment. However, we feel that we obtained some results to open a new direction of the research on the CNN.

References

[1] L.O. Chua and L. Yang, "Cellular Neural Networks:Theory," IEEE Trans. Circuits Syst., vol. 32, pp. 1257-1272, Oct. 1988.

[2] F. Dirk and T. Ronald, "Coding of Binary Image Data using Cellular Neural Networks and Iterative Annealing," Proc. of ECCTD'03, vol. 1, pp. 229-232, Sep. 2003.

[3] M. Namba and Z. Zhang, "Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition," Proc. of IJCNN'06, pp. 4716-4721, Jul. 2006.

[4] H. Koeppl and L.O. Chua, "An Adaptive Cellular Nonlinear Network and its Application," Proc. of NOLTA'07, pp. 15-18, Sep. 2007.

[5] T. Kozek, K.R. Crounse, T. Roska and L.O. Chua, "Smart Image Scanning Algorithms for the CNN Universal Machine," Proc. of NOLTA'95, vol. 2, pp. 707-712, 1995.

[6] J. Kishida, C. Rekeczky, Y. Nishio and A. Ushida, "Feature Extraction of Postage Stamps Using an Iterative Approach of CNN," IEICE Trans. on Fundamentals, vol. E79-A, no. 10, pp. 1741-1746, Oct. 1996.

[7] 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.

[8] 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.

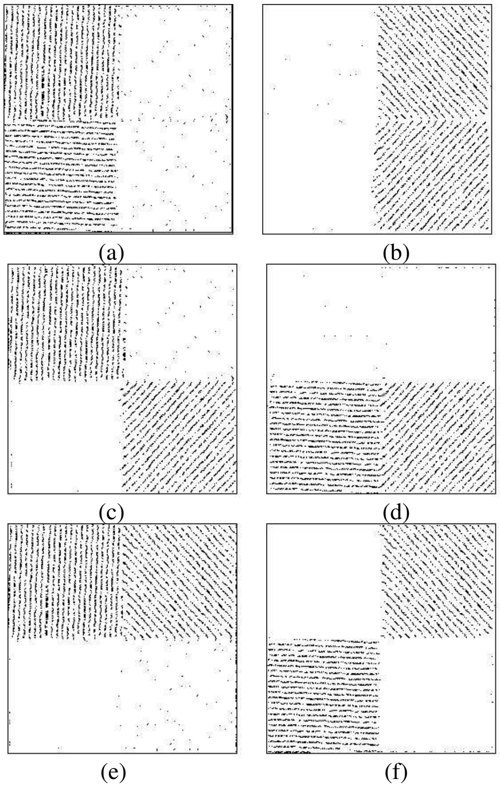


Fig. 14. Simulation results for the input image in Fig. 12(b) using mixture template with matrix mixing. (a) Vertical Line Detecter : Horizon Line Detect=50:50. (b) Diagonally Right Up Line Detect : Diagonally Left Up Line Detect=50:50. (c) Diagonally Right Up Line Detect : Vertical Line Detect=50:50. (d) Horizon Line Detect : Diagonally Right Up Line Detect=50:50. (e) Vertical Line Detect : Diagonally Left Up Line Detect=50:50. (f) Horzon Line Detect : Deiagonally Left Up Line Detect=50:50.

[9] Taisuke NISHIO and Yoshifumi NISHIO, "Image Processing Using Periodic Pattern Formation in Cellular Neural Networks," Proceedings of European Conference on Circuit Theory and Design (ECCTD'05), vol. 3, pp. 85-88, Aug. 2005.