2010 IEEE Workshop on
Nonlinear Circuit Networks, Tokushima, Dec 3-4, 2010

Output Characteristics of CNN with Dynamic Template for Motion Pictures

Masakazu KAWAHARA
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: kawahara@ee.tokushima-u.ac.jp

Yoko UWATE
Dept. Electrical and Electronic Eng.,
Tokushima University
uwate@ee.tokushima-u.ac.jp

Yoshifumi NISHIO
Dept. Electrical and Electronic Eng.,
Tokushima University
nishio@ee.tokushima-u.ac.jp

***Abstract*—In our previous research, cellular neural networks with dynamic template (D-CNN) have been proposed. In D-CNN, the wiring weights of template are dynamically changed at each update by learning. In this study, we investigate output characteristic in different two types definition of updated process. And we investigate output characteristics and compare the proposed method with the previous update method.**

I. Introduction

Cellular Neural Networks (CNN) were proposed by Chua and Yang in 1988 [1]. The idea of CNN was inspired from the architecture of the cellular automata and the neural networks. Unlike the original neural networks, the CNN has local connectivity property. Wiring weights of the cells are established by parameters called the template. The performance of the CNN is decided by the template. Also, the CNN has been successfully used for various high-speed parallel signals processing applications such as image processing application [2]-[4]. Usually, the templates of all the cells in the CNN are identical and those values do not change during the processing. This is good for implementation however restrict the performance, namely the conventional CNN can not perform image processing based on the local features of input images.

In the previous study, we have proposed CNN with dynamic template (D-CNN) [5]. In D-CNN, template is dynamically changed at each update by learning. This learning method is inspired from the rank order learning. The updated template depends on the output value of cells. From the simulation results of the previous study, we have confirmed that the converged value of each cell is divided to two or three values. Also, convergence process is much more rapid than that of the conventional CNN. Then, we have investigated update template in D-CNN for motion pictures [6][7].

In this study, we investigate output characteristics of D-CNN for motion pictures by changing updated process. In our proposed method, the templates are updated at every iterations and input images are changed when certain calculation times come. By using D-CNN, we carry out moving object detection from motion pictures. From the simulation results, we confirm the effectivity of moving object detection using D-CNN. The rest of this paper is structured as follows. In the Sec. 2, we review the basic of the standard CNN. In the Sec. 3, we show the algorithm of the proposed D-CNN. In the Sec. 4, simulation results using the proposed D-CNN are shown. The Section 5 concludes the article.

II. Cellular Neural Networks [1]

In this section, we explain the basic structure of the CNN. The CNN has M by N processing unit circuits called cells. 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.

State equation:

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

Output equation:

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

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

III. CNN with Dynamic Template [2]

In this section, we explain the algorithm of D-CNN. In our research, we change input images when a certain calculation times comes. In our D-CNN, the templates are updated at every iterations by rank order learning. Also, in this study we defined two types updated methods in D-CNN. The learning steps in our D-CNN are described as follows.

STEP 1: The state values and the output values of all the cells in D-CNN are updated according to the discretized model of Eqs. (1) and (2).

STEP 2: Calculate the comparison of the output value of each cell with the one-step-past outputs of the cell and its neighbor cells. The comparison equation for the cell (i, j) is described Eq. (3).

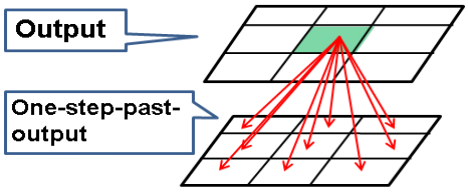

Fig. 1. Comparison of output value of each cell with one-step-past outputs of the cell and its neighbor cells.

Comparison Equation:

$$Dif(i, j; k, l) = |v^{past}_{y,(i,j)} - v^{now}_{y,(k,l)}|. \tag{3}$$

STEP 3: Among the 9 calculated values of $Dif(i, j; k, l)$, the cells with the most smallest and the second smallest values are defined as "winner" and "second", respectively. In our update algorithm, we change the learning rate in two elements. By this step, we find the position of cells with the nearest and the second nearest values to the corresponding cell (i, j).

In this study, we change the decision method of "winner" and "second". In the previous study, the template of D-CNN is updated even if all the value of $Dif(i, j; k, l)$ are same value (Type1). On the other hand, if all the value of $Dif(i, j; k, l)$ are same value, template is not updated in this study (Type2). We investigate the difference between these two kinds of updated processes.

STEP 4: Update the elements of the template corresponding to the positions of the "winner" and the "second". Note that in our proposed learning algorithm only two elements are updated. The update method and the update function are described as follows.

Update Method:

Assume that the template before update is given as Eq. (4).

$Template^{now}$:

$$A^{now}_{(i,j)} = \begin{bmatrix} a^{now}_{11} & a^{now}_{12} & a^{now}_{13} \\ a^{now}_{21} & a^{now}_{22} & a^{now}_{23} \\ a^{now}_{31} & a^{now}_{32} & a^{now}_{33} \end{bmatrix},$$
$$B^{now}_{(i,j)} = \begin{bmatrix} b^{now}_{11} & b^{now}_{12} & b^{now}_{13} \\ b^{now}_{21} & b^{now}_{22} & b^{now}_{23} \\ b^{now}_{31} & b^{now}_{32} & b^{now}_{33} \end{bmatrix},$$
$$I^{now}_{(i,j)} = I^{now}. \tag{4}$$

For example, we consider the case that the "winner" is $C_{(i,j)}$ and the "second" is $C_{(i-1,j-1)}$. In that case, only a^{now}_{22}, b^{now}_{22}, a^{now}_{11} and b^{now}_{11} in Eq. (4) are updated. The threshold value I is not updated in our learning method.

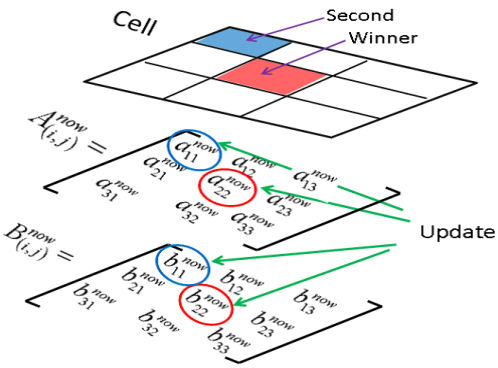

Fig. 2. Decision of updated elements of template in D-CNN.

In our update algorithm, we change the learning rate in two elements. The learning rates of the "winner" and the "second" are shown as follows.

Learning rate:

$$R_1 = R_{10}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \tag{5}$$
$$R_2 = R_{20}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \tag{6}$$

In this study, we decide $Number\ of\ calculation_{max}$ in Eqs. (5) and (6) to be set to 10. Namely, the learning rates of "winner" and "second" are changed until 10 calculations. Then, after $Number\ of\ calculation_{max}$ becomes over 10, the learning rates of "winner" and "second" become 0 and the templates are not updated. By using the learning rate, the elements of the template are updated according to the following update equation.

Update Equation:

$$a^{updated}_{winner} = a^{now}_{winner} - R_1(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{7}$$
$$a^{updated}_{second} = a^{now}_{second} - R_2(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{8}$$

R_1 and R_2 decrease according to the Eqs. (5) and (6). The initial learning rates are given as follows.

Initial Learning rate:

$$Winner \quad : \quad R_{10} \quad (0 \le R_{10} \le 0.1). \tag{9}$$
$$Second \quad : \quad R_{20} = R_{10}/4. \tag{10}$$

After the update using Eqs. (7) and (8), the updated template is shown as follows. In Eq. (11), $a^{updated}_{11}$ and $a^{updated}_{22}$ are the updated values. Also, $b^{updated}_{11}$ and $b^{updated}_{22}$ are updated similarly.

$Template^{updated}$:

$$A^{updated}_{(i,j)} = \begin{bmatrix} a^{updated}_{11} & a^{now}_{12} & a^{now}_{13} \\ a^{now}_{21} & a^{updated}_{22} & a^{now}_{23} \\ a^{now}_{31} & a^{now}_{32} & a^{now}_{33} \end{bmatrix},$$
$$B^{updated}_{(i,j)} = \begin{bmatrix} b^{updated}_{11} & b^{now}_{12} & b^{now}_{13} \\ b^{now}_{21} & b^{updated}_{22} & b^{now}_{23} \\ b^{now}_{31} & b^{now}_{32} & b^{now}_{33} \end{bmatrix},$$

$$I_{(i,j)}^{updated} = I^{now}. \quad (11)$$

STEP 5: The steps from 1 to 4 are repeated.

These learning steps inspired from the rank order learning.

IV. Simulation Results

In this section, we show the simulation results for two types of motion pictures. We investigate the difference between "Type1" and "Type2" in updated process of D-CNN. We also compare conventional CNN with D-CNN. In the first step of this investigation, an initial template is set to D-CNN. The elements of the initial template are updated by using updated method in the previous section. In motion picture processing, the template of D-CNN updated every input image changed. The initial template used in this study is described as follows.

Initial Template:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.07 & 0.1 & 0.07 \\ 0.1 & 0.32 & 0.1 \\ 0.07 & 0.1 & 0.07 \end{bmatrix}, I = 0. \quad (12)$$

In this study, R_{10} and R_{20} of "winner" and "second" set to 0.1 and 0.025.

A. Motion picture 1

Firstly, we investigate updated template for simple binary image. We change 4 input images every 10 $[\tau]$ like a motion picture.

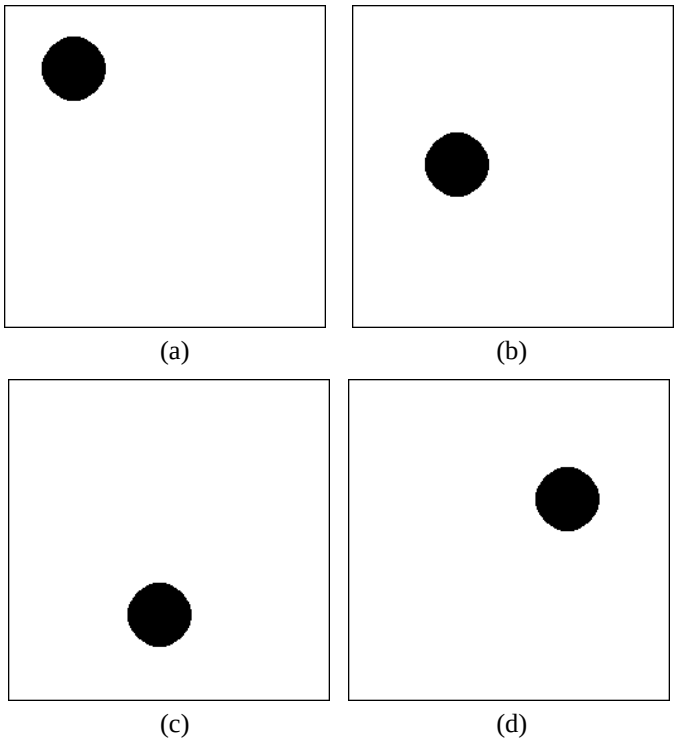

Fig. 3. Motion picture 1. (a) Input image 1 (from 0 to 10 $[\tau]$). (b) Input image 2 (from 10 to 20 $[\tau]$). (c) Input image 3 (from 20 to 30 $[\tau]$). (d) Input image 4 (after 30 $[\tau]$).

Figure 3 shows binary input images as motion pictures. We recognize the black object moved left to right through Figs. 3(a) to (d). By changing these input images, we compare the output characteristics of "Type1" with that of "Type2".

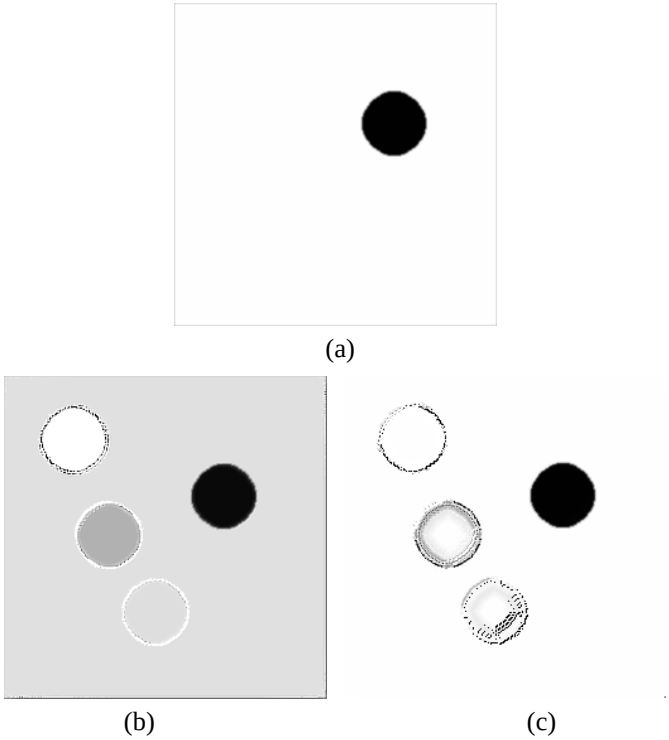

Fig. 4. The simulation results for motion picture1. (a) Output image using conventional CNN. (b) Output image using D-CNN (Type1). (c) Output image using D-CNN (Type2).

Figure 4 shows the simulation results for motion pictures. Figure 4(a) shows the output image using conventional CNN. The present image is only influenced and the previous image is not influenced. Namely, conventional CNN is not suitable motion picture processing. On the other hand, all the images are influenced by using D-CNN in "Type1" as shown in Fig. 4(b). This result is the effect of the learning of template. However, the value of background is changed and we cannot recognize the trace of moving object. In Fig. 4(c) we can recognize the trace of moving object by using "Type2". Also, the edge of all the image can be detected by using D-CNN. From these simulation results we can say that our proposed D-CNN in "Type2" is more effective motion picture processing than that of "Type1".

B. Motion picture 2

Next, we investigate more realistic gray scale images for a motion picture. Similar to the previous one, we also change 4 input images every 10 $[\tau]$ like a motion picture.

Figure 5 shows the input images. We recognize a person gradually moved from right to left through Figs. 5(a) to (d). The background almost does not change among these images. Using these input images, we investigate the output characteristic of D-CNN.

Figure 6 shows the simulation result with the input images of Fig. 5. Similar to the case of the simple figures in the

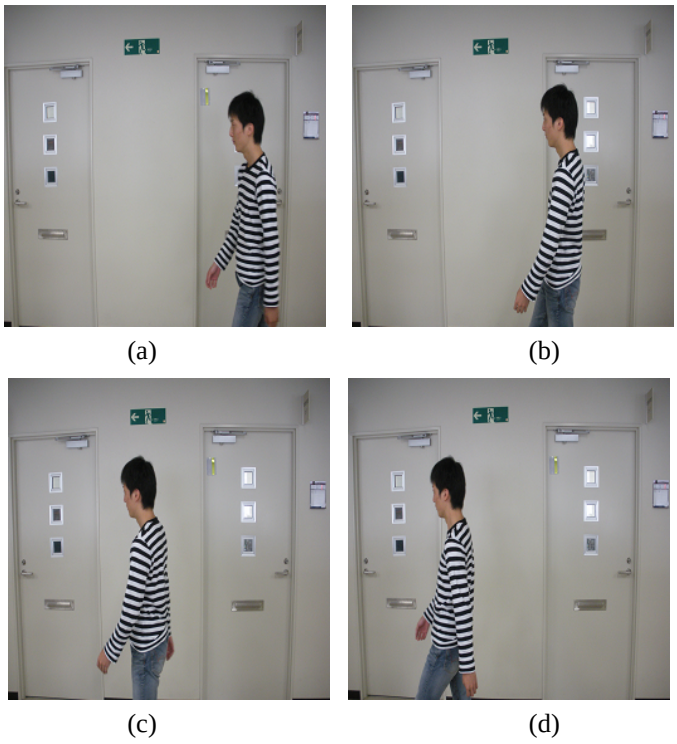

Fig. 5. Motion picture 2. (a) Input image 1 (from 0 to 10 $[\tau]$). (b) Input image 2 (from 10 to 20 $[\tau]$). (c) Input image 3 (from 20 to 30 $[\tau]$). (d) Input image 4 (after 30 $[\tau]$).

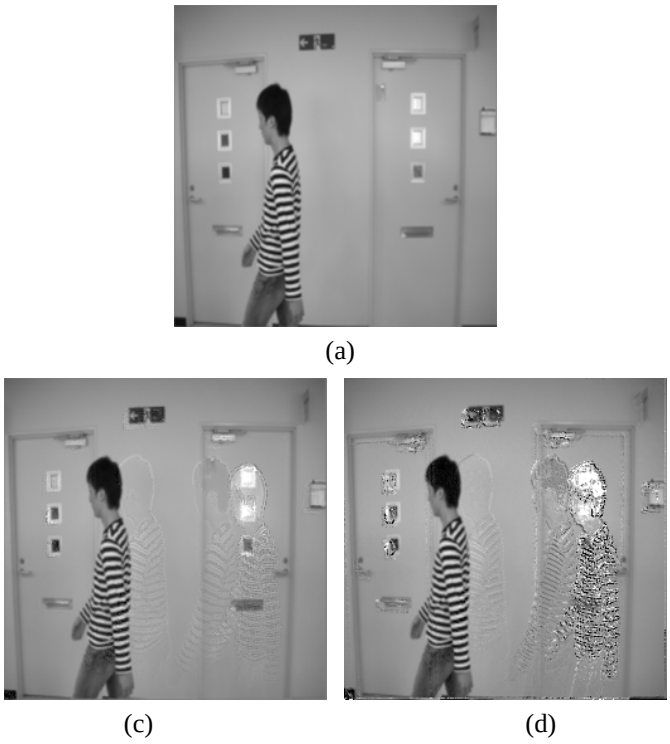

Fig. 6. The simulation results for motion picture2. (a) Output image using conventional CNN for the input converted to gray-scale image. (b) Output image using D-CNN (Type1) for the input converted to gray-scale image. (c) Output image using D-CNN (Type2) for the input converted to gray-scale image.

above example, the proposed D-CNN successfully detects the moving object as a edged shape like a ghost overlapped with the present input image. However, the edge of moving object is very thin in "Type1". On the other hand, the detected edge in "Type2" is more clearly than that of "Type1". Also, the final steady state output includes all the edges indicating the moving parts of the images with the final input image.

V. Conclusions

In this study, we have investigated the characteristics of the output characteristics of D-CNN when input images were changed during D-CNN process. In D-CNN, the template was changed by rank order learning. Also, we have investigated output characteristics of some different definition types as updated process in D-CNN. From the simulation results for two types of input motion pictures, we succeeded at moving object detection than the previous study. Similar results might be obtained by using CNN with delay-type template [8]. However, we consider that the proposed method shows a new motion picture processing which takes an advantage of the rank order learning as the characteristic of D-CNN. Realizing more intelligent tasks depending on inputs is our future research.

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] M. Kawahara, T. Inoue and Y. Nishio, "Cellular Neural Network with Dynamic Template and its Output Characteristics," Proc. of IJCNN'09, pp. 1552-1558, Jun. 2009
[6] M. Kawahara, T. Inoue and Y. Nishio, "Image Processing Application Using CNN with Dynamic Template" Proc. of CNNA'10, pp. 41-46, Feb. 2010.
[7] M. Kawahara, Y. Uwate and Y. Nishio, "Characteristics of Cellular Neural Networks with Dynamic Template for Motion Pictures" Proc. of NOLTA'10, pp. 23-26, Sep. 2010.
[8] T. Roska, T. Boros A. Radvanyi P. Thiran and L. O. Chua, "Cellular Neural Networks with Non-Linear and Delay-Type Template Elements and Non-Uniform," Internationl Journal of Circuit Theory and Appliacations, vol. 20, pp. 469-481, Sept.-Oct. 1992.