

Four-Layer Cellular Neural Networks in Consideration of Color and Luminosity

Yoshihiro Kato, Yasuhiro Ueda, Yoko Uwate and Yoshifumi Nishio
 Department of Electrical and Electronic Engineering, Tokushima University,
 2-1 Minami Josanjima, Tokushima 770-8506, Japan
 Email: {kkato, yasuhiko, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—Human’s retina has the capability to identify color and luminosity. The cell identifies color is called a cone cell and identifies luminosity is called a rod cell. The color image processing using CNN was proposed by Roska et al. Additionally, Inoue et al. have used three-layer CNN based on cone cell, and performed edge enhancement. They have confirmed that edge had been detected under three-layer CNN influencing each other. However, the edge of a low luminosity portion is not detected. In this study, we propose four-layer cellular neural networks in consideration of three primary colors of light and luminosity, respectively. In this research, we show some edge detection results and confirm that the proposed CNN is effective compared with the conventional CNN and the existing CNN.

I. INTRODUCTION

In recent years, our life teems with information by growth of high information society. Generally, digital circuit is used for many information processing. However, digital circuit cannot perform real time processing. Therefore, the neural networks were devised based on the human’s nervous system. The neural networks can design of analog circuit. Additionally, the neural networks can perform expression of nonlinear operating characteristics. Hence, the neural networks can express complicated processing easily. In fact, neural networks are used in every direction, such as an associative memory, pattern recognition, image processing, and so on. Then in 1988, Cellular Neural Networks (CNN) [1] were introduced by L. O. Chua and L. Yang. The idea of the CNN was inspired from the architecture of the cellular automata and the neural networks. A different point from the conventional neural networks is that CNN has local connectivity property. Since, the structure of CNN resembles that of animals’ retina. Therefore, CNN can be used for various image processing applications [2]-[5] including pattern formation [6].

Human’s retina has the capability to identify color and luminosity. The cell identifies color is called a cone cell. Cone cell consists of cells which react to the three primary colors of light which are red, green and blue, respectively. Cone cell often functions in bright place. However, cone cell hardly functions in dark place. On the other hand, the cell reacting to luminosity is called a rod cell. Rod cell often functions in dark place. However, rod cell cannot identify a color. The color image processing using CNN was proposed by Roska et al [7]. Additionally, Inoue et al. have used three-layer CNN based on cone cell which processes the three primary colors of light (R, G, B), and performed edge enhancement in [8]. The three-layer

CNN is called RGB-CNN. They have confirmed that edge had been detected under three-layer CNN influencing each other. However, the edge of a low luminosity portion is not detected.

In this study, we propose the four-layer cellular neural networks considering cone cell and rod cell which process the three primary colors of light and the luminosity (CR-CNN: Cone and Rod CNN). Before the processing, a color image is converted to four gray-scale images. They correspond to red value, green value, blue value and luminosity value, respectively. This preprocessing corresponds to the function of the cone and rod cells of the human’s retina. These four gray-scale images are inputted to each layer in the proposed four-layer CNN.

The rest of this paper is structured as follows. In Sec. II, we propose the CR-CNN. In Sec. III, we show some simulation results. The section IV concludes the article.

II. CR-CNN

A. Cellular Neural Networks

The CNN has M by N processing unit circuits called cells. The cell contains linear and nonlinear circuit elements which are typically linear capacitors, linear resistors, linear and nonlinear controlled sources. 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 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. Additionally, the local connectivity property makes the CNN tailor made for VLSI implementation.

B. Structure of the CR-CNN

In this subsection, we explain the structure of the CR-CNN as shown in Fig. 1. The CR-CNN is constructed by four single-layer CNNs. Each layer corresponds to the three primary colors of light which are red, green and blue and luminosity, respectively. Each layer is combined by connection template which are C_R , C_G , C_B , C_L , C_{LR} , C_{LG} and C_{LB} . The state equation and the output equation of each layer are shown as follows.

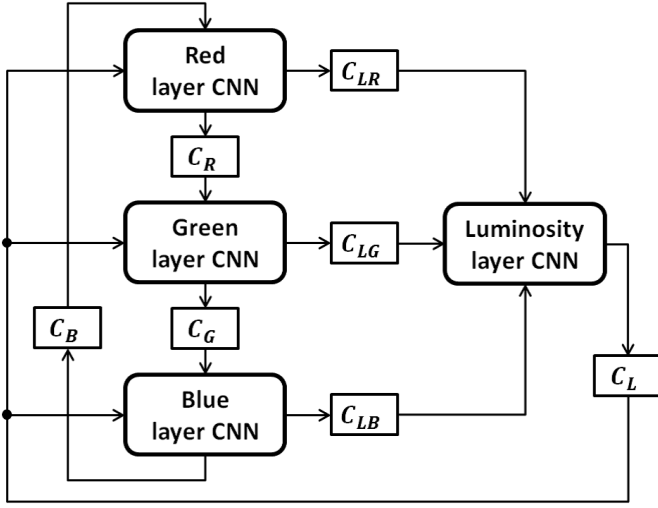


Fig. 1. Structure of the CR-CNN.

State equation of red-layer :

$$\begin{aligned}
\frac{dv_{xRij}}{dt} &= -v_{xRij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{R(i,j;k,l)} v_{yRkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{R(i,j;k,l)} v_{uRkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{B(i,j;k,l)} v_{yBkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{L(i,j;k,l)} v_{yLkl}(t) + I_R \quad (1)
\end{aligned}$$

State equation of green-layer :

$$\begin{aligned}
\frac{dv_{xGij}}{dt} &= -v_{xGij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{G(i,j;k,l)} v_{yGkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{G(i,j;k,l)} v_{uGkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{R(i,j;k,l)} v_{yRkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{L(i,j;k,l)} v_{yLkl}(t) + I_G \quad (2)
\end{aligned}$$

State equation of blue-layer :

$$\begin{aligned}
\frac{dv_{xBij}}{dt} &= -v_{xBij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{B(i,j;k,l)} v_{yBkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{B(i,j;k,l)} v_{uBkl}(t)
\end{aligned}$$

$$\begin{aligned}
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{G(i,j;k,l)} v_{yGkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{L(i,j;k,l)} v_{yLkl}(t) + I_B \quad (3)
\end{aligned}$$

State equation of luminosity-layer :

$$\begin{aligned}
\frac{dv_{xLij}}{dt} &= -v_{xLij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{L(i,j;k,l)} v_{yLkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{L(i,j;k,l)} v_{uLkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{LR(i,j;k,l)} v_{yRkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{LG(i,j;k,l)} v_{yGkl}(t) \\
&+ \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} C_{LB(i,j;k,l)} v_{yBkl}(t) + I_L \quad (4)
\end{aligned}$$

Output equation :

$$v_{yRij}(t) = \frac{1}{2}(|v_{xRij}(t) + 1| - |v_{xRij}(t) - 1|) \quad (5)$$

$$v_{yGij}(t) = \frac{1}{2}(|v_{xGij}(t) + 1| - |v_{xGij}(t) - 1|) \quad (6)$$

$$v_{yBij}(t) = \frac{1}{2}(|v_{xBij}(t) + 1| - |v_{xBij}(t) - 1|) \quad (7)$$

$$v_{yLij}(t) = \frac{1}{2}(|v_{xLij}(t) + 1| - |v_{xLij}(t) - 1|) \quad (8)$$

where v_x , v_y and v_u represent a state, an output and an input value of cell, respectively. In the each state equation, A is the feedback template, B is the control template, C is the connection template from other layer and I is the constant bias. The output equation has the feature of piece-wise linear function. In fact, the output value of the CR-CNN is within of -1 to 1 .

III. SIMULATION RESULTS

In this section, we show some simulation results for edge detection by using the CR-CNN. For this simulation, we use some general color images. Figure 2 shows an example of the dissolution of a color image to four gray-scale images. Figure 2 (a) is the original color image. Also Figs. 2 (b), (c), (d) and (e) are the gray-scale images corresponding to the luminosity, red, green and blue values of the color image, respectively. They are used as input image of each layer, respectively.

Templates for edge detection by using the CR-CNN are shown as follows.



(a)



(b)



(c)



(d)



(e)

Fig. 2. Gray-scale image. (a) Original color image. (b) Gray-scale image for luminosity value. (c) Gray-scale image for red value. (d) Gray-scale image for green value. (e) Gray-scale image for blue value.

Edge detection template in the CR-CNN :

$$A_R = A_G = A_B = A_L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_R = B_G = B_B = B_L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$C_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yRij} \geq 0 \\ b = 0 & \text{otherwise} \end{cases}$$

$$C_G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yGij} \geq 0 \\ b = 0 & \text{otherwise} \end{cases}$$

$$C_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yBij} \geq 0 \\ b = 0 & \text{otherwise} \end{cases}$$

$$C_L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 1 & \text{if } v_{yLij} \geq 0 \\ b = 0 & \text{otherwise} \end{cases}$$

$$C_{LR} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yRij} \geq 0, \\ & v_{yRij} > v_{yGij} \text{ and} \\ & v_{yRij} > v_{yBij} \\ b = 0 & \text{otherwise} \end{cases}$$

$$C_{LG} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yGij} \geq 0, \\ & v_{yGij} > v_{yRij} \text{ and} \\ & v_{yGij} > v_{yBij} \\ b = 0 & \text{otherwise} \end{cases}$$

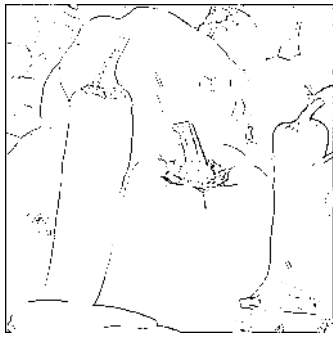
$$C_{LB} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{cases} b = 0.9 & \text{if } v_{yBij} \geq 0, \\ & v_{yBij} > v_{yRij} \text{ and} \\ & v_{yBij} > v_{yGij} \\ b = 0 & \text{otherwise} \end{cases}$$

$$I_R = I_G = I_B = I_L = -1, \quad (9)$$

where A , B and I of each layer are the same as existing edge detection template in [9]. Red, green and blue layers are influenced through connection template (C_B , C_R , C_G and C_L) from blue, red and green layers, respectively and luminosity layer. Luminosity layer is influenced through connection template (C_{LR} , C_{LG} and C_{LB}) only from the cell whose value is the large among red, green and blue layer when the value is larger than zero. In the case of each cell are the same output value, luminosity layer is influenced through connection template only from one of them. In this simulation result, the output images of all the layers became almost the same image.

A. Pepper Image

Figure 3 shows the simulation results of edge detection. The input images of each layer are Figs. 2 (b) to (e). In Fig. 3 (b), using the RGB-CNN, edge is more detected than the conventional CNN like Fig. 3 (a). However, the edge of a low luminosity portion remains as black. In Fig. 3 (c), we can see that the CR-CNN can detect the edges more effectively than the conventional CNN and the RGB-CNN.



(a)



(b)

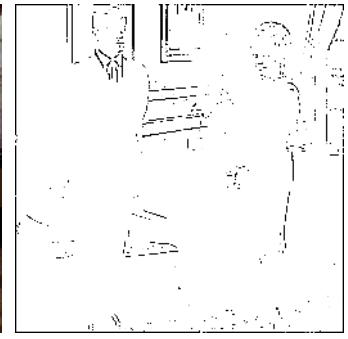


(c)

Fig. 3. Simulation results 1. (a) Edge detection result using the conventional CNN for the input image converted to gray-scale image. (b) Edge detection result using the RGB-CNN. (c) Edge detection result using the CR-CNN (output image of blue layer).



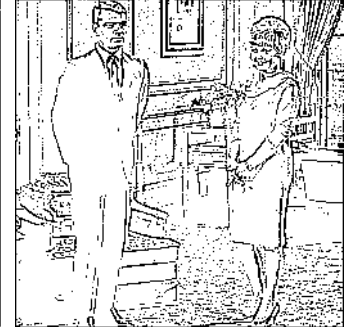
(a)



(b)



(c)



(d)

Fig. 4. Simulation results 2. (a) Input image. (b) Edge detection result using the conventional CNN for the input image converted to gray-scale image. (c) Edge detection result using the RGB-CNN. (d) Edge detection result using the CR-CNN (output image of blue layer).

B. Couple Image

Figure 4 shows the simulation results of edge detection. In Fig. 4 (a), this input image is a low luminosity image. In Fig. 4 (b), using the conventional CNN, most edge is not detected. In Fig. 4 (c), using the RGB-CNN, this output image is not edge detection image which would appear that the input image is a low luminosity. In Fig. 4 (d), we can see that the CR-CNN can detect the edges more effectively than the conventional CNN and the RGB-CNN.

The simulation results by using the CR-CNN arise from the structure which consideration of three primary colors of light and luminosity.

IV. CONCLUSION

In this study, we have proposed four-layer CNN in consideration of three primary colors of light and luminosity. From some simulation results, we have confirmed that the CR-CNN could detect the edge of color images more effectively than the conventional CNN and the RGB-CNN. In the future work, we would like to use the CR-CNN for other task of color image processing.

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. Koepl 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. Crouse, 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] K. R. Crouse 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.

[7] T. Roska, A. Zarandy and L. O. Chua, "Color Image Processing by CNN," *Proc. of ECCTD'93*, pp. 57-62, Aug. 1993.

[8] T. Inoue and Y. Nishio, "Edge Enhancement of Color Image by Three-Layer Cellular Neural Network Considering Three Primary Colors," *Proc. of NOLTA'08*, pp. 540-543, Sep. 2008.

[9] Cellular Sensory Wave Computers Laboratory Computer and Automation Research Institute Hungarian Academy of Sciences, "Cellular wave Computing Library (Template, Algorithms, and Programs) Version 2.1"