2020 RISP International Workshop on Nonlinear Circuits,
Communications and Signal Processing (NCSP 2020)
Honolulu, Hawaii, USA, February 28 - March 2, 2020

Image Sharpening for Blurred Images with Multiple Processes by Using Cellular Neural Networks

Kodai Kitamura, Yoko Uwate and Yoshifumi Nishio

Department of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima 770–8506, Japan
E-mail: {kitamura, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract

The Cellular Neural Networks (CNN) was developed by Chua and Yang in 1998. The idea of the CNN was inspired from the architecture of the cellular automata and the neural networks. The performance of the CNN depends on the parameters which are called the template. The CNN is applied to various image processing by changing the template. However, little has been reported on image sharpening for blurred images by using CNN. In this study, we propose an algorithm to sharpen blurred images. Then, we apply the proposed algorithm to the input image and investigate the performance by some simulations.

1. INTRODUCTION

In recent years, it has advanced information society. In general, the von Neumann computer is used for many information processing. However, the von Neumann computer cannot process many information in real time. Therefore, Neural Networks were proposed. The idea was based on the human's nervous system. The Cellular Neural Networks (CNN) was introduced by L. O. Chua and L. Yang in 1988 [1]. Hence, the CNN has been successfully used for various high-speed parallel signals processing applications such as image processing, pattern recognition and so on [1]. The CNN was paired Neural Networks with the Cellular Automata by L. O. Chua in 1998 [2]. The CNN consists of cells connected each other and the structure of CNN resembles the structure of the animal's retina [3]. Therefore, the CNN has been applied to various image processing applications. The performance of CNN depends on the parameters which are called the template. The template represents strength of connection between each cell. The template consists of the feedback template, the feedforward template and the threshold. If the template influences exactly, we can investigate complex image processing. Additionally, various applications for image processing and pattern recognition of CNN have been reported.

In previous method, there is the Unsharp Masking (USM). The USM is used as digital image processing to sharpen images. In the USM, first, the input image is smoothed. Secondly, a difference image between the input image and the smooth image is created. Finally, the output image is a combination of the input image and the difference image. However, little image sharpening for blurred images by using CNN has been reported.

In this study, we propose an algorithm to sharpen images by using CNN. Our proposed method is performed on blurred images. In order to confirm the effectiveness of our proposed method, we evaluate simulation results.

2. CELLULAR NEURAL NETWORKS

In this section, we describe the basic structure and processing flow of the simple CNN. The basic circuit unit of the CNN is called a cell. The cell consists of linear element and nonlinear element. The CNN is formed from an array of many cells. We show a two dimensional array composed of $M \times N$ cells arranged in M rows and N columes. The structure of the CNN is shown in Fig. 1. Figure 2 shows the circuit of cell.

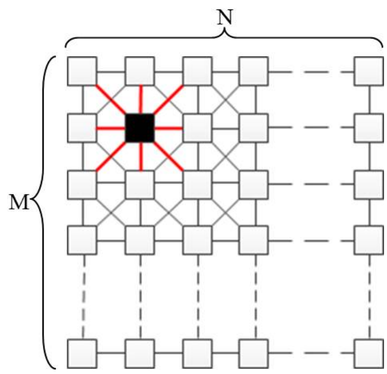

Figure 1: The structure of CNN.

A cell is coupled with only adjacent cells. Adjacent cells interact with one another. Cells which do not couple with only adjacent cells have an indirect influence. The range which some cells influence one cell is defined by neighborhood. We describe a state equation of cell and an output equation of cell

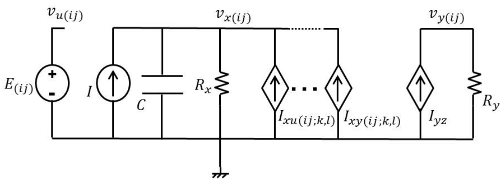

Figure 2: The circuit of cell.

below.

State Equation :

$$\frac{dv_{x(ij)}}{dt} = -v_{x(ij)} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{(i,j;k,l)} v_{y(kl)}(t) + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{(i,j;k,l)} v_{u(kl)}(t) + T. \quad (1)$$

Output Equation :

$$v_{y(ij)}(t) = \frac{1}{2}(|v_{x(ij)}(t) + 1| - |v_{x(ij)}(t) - 1|). \quad (2)$$

v_x, v_y and v_u are state value, output value and input value. Output equation represents piece-wise nonlinear function Fig 3.

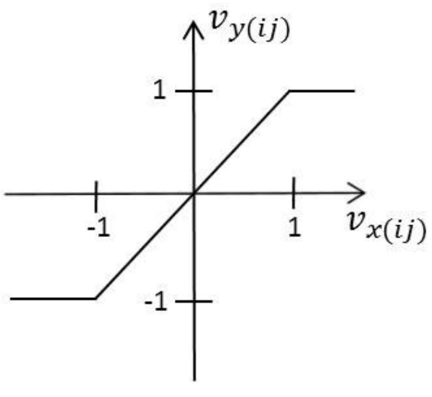

Figure 3: Piece-wise nonlinear function.

In Eq. (1), A is feedback template, B is feedforward template, T is threshold. The performance of CNN is determined by these values.

When we process the image by using CNN, we should determine the size of system of neighborhood. When this size is big, the amount of information in system of neighborhood is increasing. However, noise works easily. Generally, the size which is used for image processing is 3×3. Figure 4 shows block diagram of image processing with CNN.

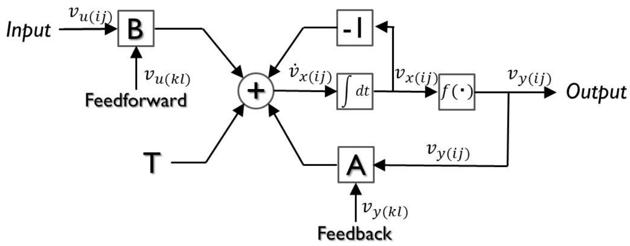

Figure 4: Block diagram.

3. PROPOSED METHOD

In this section, we show the proposed algorithm by using CNN. The proposed algorithm is composed of six steps and the flowchart is described as follows.

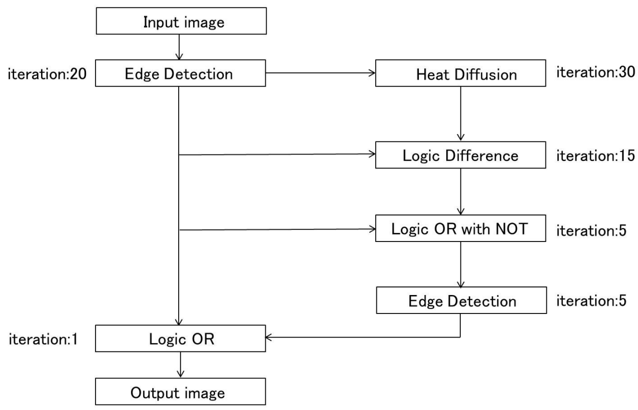

Figure 5: The proposed algorithm.

The processing steps of the proposed method are described as follows.

Step 1 (Edge Detection): First, the input image is sharpened.

Step 2 (Heat Diffusion): Second, the sharpened image is blurred to subtract the heat diffused image from the sharpened image.

Step 3 (Logic Difference): Third, sharp lines of the sharpened image are detected.

Step 4 (Logic OR with NOT): Fourth, sharp lines that cannot be detected in Logic Difference are detected.

Step 5 (Edge Detection): Fifth, sharp lines of a previous step are enhanced.

Step 6 (Logic OR): Final, each cell's output value of the sharpened image and the enhanced edge detected image is added. Each step size is 0.05 [τ].

4. SIMULATION RESULTS

In this section, we show the simulation results of the proposed algorithm and the USM. Figure 2 shows how the image is processed in the proposed algorithm and the USM. Using templates of the proposed algorithm are described as follows [5].

Heat Diffusion template:

$$A = \begin{bmatrix} 0.1 & 0.15 & 0.1 \\ 0.15 & 0 & 0.15 \\ 0.1 & 0.15 & 0.1 \end{bmatrix},$$
$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, T = 0. \quad (3)$$

Edge Detection template:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, T = 0. \quad (4)$$

Logic Difference template:

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

Logic OR with NOT template:

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

Logic OR template:

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

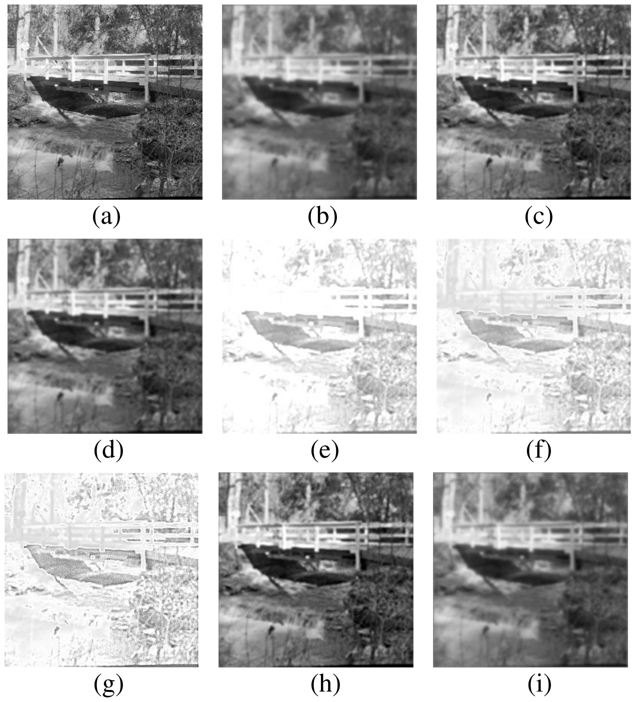

Figure 6: Simulation results. (a) Original image. (b) Input image. (c) Sharpened image. (d) Heat diffused image. (e) Logic difference. (f) Logic OR with NOT. (g) Enhanced edge detected image. (h) Output image. (i) Unsharp masked image.

In one instance, Fig. 6 show images which are processed by the proposed method and the USM. Figure 6(a) shows an original image. The input image contains a bridge, a lake and a background of the forest. Figure 6(b) shows the simulation result of Heat Diffusion applied to the input image by iteration of 100. The overall image is blurred. Figure 6(c) shows the simulation result of Edge Detection applied to the input image. The overall image is sharpened compared to Fig. 6(c). Figure 6(d) shows the simulation result of Heat Diffusion applied to the sharpened image. The overall image is blurred compared to Fig. 6(c). Figure 6(e) shows the simulation result of Logic Difference applied to the sharpened image and the heat diffused image. The enhanced sharp lines of a lake and a background of the forest are detected. Figure 6(f) shows the simulation result of Logic OR with NOT applied to the sharpened image and the heat diffused image. The left side of the forest which was not detected in Fig. 6(e) was detected. Figure 6(g) shows the simulation result of Edge Detection applied to the enhanced image. The enhanced sharp lines of the sharpened image are detected compared to Fig. 6(g). Figure 6(h) shows the simulation result of Logic OR applied to the sharpened image and the enhanced image. Not many unclear objects are remained. On the other hand, in Fig. 6(i), unclear objects are more remained compared to Fig. 6(h).

We process another image which is processed by the proposed method and the USM Figure 7(a) shows an original im-

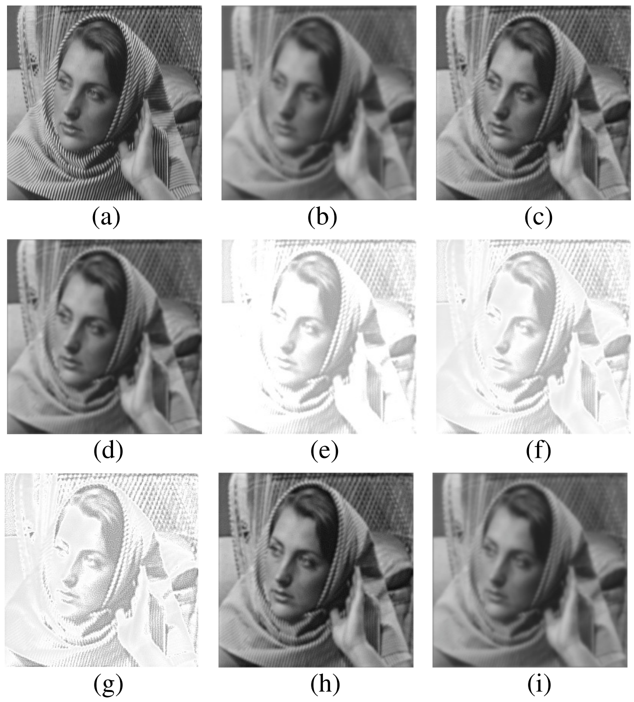

Figure 7: Simulation results 2. (a) Original image. (b) Input image. (c) Sharpened image. (d) Heat diffused image. (e) Logic difference. (f) Logic OR with NOT. (g) Enhanced edge detected image. (h) Output image. (i) Unsharp masked image.

age. The input image contains a woman and a reticulate background. Figure 7(b) shows the simulation result of Heat Diffusion applied to the input image by iteration of 100. The overall image is blurred. Figure 7(c) shows the simulation result of Edge Detection applied to the input image. The overall image is more sharpened compared to Fig. 7(c). Figure 7(d) shows the simulation result of Heat Diffusion applied to the sharpened image. The overall image is more blurred compared to Fig. 7(c). Figure 7(e) shows the simulation result of Logic Difference applied to the sharpened image and the heat diffused image. The enhanced sharp lines of a reticulate background are detected. Figure 7(f) shows the simulation result of Logic OR with NOT applied to the sharpened image and the heat diffused image. The left side of the forest that was not detected in Fig. 7(e) was detected. Figure 7(g) shows the simulation result of Edge Detection applied to the enhanced image. The enhanced sharp lines of the sharpened image are detected compared to Fig. 7(g). Figure 7(h) shows the simulation result of Logic OR applied to the sharpened image and the enhanced image. Not many unclear objects are remained. On the other hand, in Fig. 7(i), unclear objects are more remained compared to Fig. 7(h). We have evaluated the simulation results by the Structural Similarity(SSIM). The SSIM evaluates the change of pixel values, constant, and structure between the target images. The value of the SSIM closer to 1 is similar to the target image. In Tab. 1, we show the SSIM of each output image.

Table 1: Structural Similarity

Figure 6(h)	0.6388
Figure 6(i)	0.4833
Figure 7(h)	0.7276
Figure 7(i)	0.6053

Figure 6(h)(i) are compared to Figure 6(a) and Figure 7(h)(i) are compared to Figure 7(a) for similarities. Figure 6(h) and Figure 7(h) are processed by using the proposed method. Figure 6(i) and Figure 7(i) are processed by using USM. From Tab. 1, the SSIM by using the proposed method is higher than by using USM. From these simulation results, the proposed method is more effective than USM.

5. CONCLUSION

In this study, we have proposed an algorithm for sharpening blurred images by using CNN. Moreover, we have investigated blurred images by using the USM. Then, we have evaluated similarities from the simulation results. As a result, our proposed method is more effective than the USM. In the future works, we would like to investigate the performance of another image.

References

[1] Z. H. Yang, Y. Nishio and A. Ushida,"Image processing of twolayer CNNs applications and stability," IEICE Trans. Fundamentals, vol. E85-A, no. 9, pp. 2052-2060, 2002.

[2] L. O. Chua and L. Yang, "Cellular Neural Networks Theory," IEEE Trans. Circuits Syst., vol. 35, pp. 1257-1272, 1988.

[3] Martin Hnggi and George S Moschytz, "Cellular Neural Networks Analysis, Design and Optimization," Kluwer Academic Publishers Norwell, pp. 148. 2000.

[4] K. Nakai and A. Ushida. "On a Design Technique of Cellular Neural Network,"IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences A Vel. J77 A No. 8. pp. 113. 1994.

[5] T. Roska, L. Kek, L. Nemes, A. Zarandy, and P. Szolgay,"CNN Software Library (Templates and Algorithms), Version 7.3," Analogical and Neural Computing Laboratory, Budapest, Hungary, 1999.