Journal of Mobile Multimedia, Vol. 6, No. 4 (2010) 293–304

FAULT TOLERANT SMALL-WORLD CELLULAR NEURAL NETWORKS FOR INTERMITTED FAULTS

KATSUYOSHI MATSUMOTO, MINORU UEHARA, HIDEKI MORI
Department of Open Information Systems Graduate School of Engineering, Toyo University
Kawagoe, Saitama 350-1109, Japan
dz080001x,uehara,mori@toyonet.toyo.ac.jp

Received August 20, 2009
Revised January 6, 2010

A Cellular Neural Network (CNN) is a neural network model linked only to neighborhoods and which is suitable for image processing, such as noise reduction and edge detection. A Small World Cellular Neural Network (SWCNN) is an extended CNN to which has been added a small world link, which is a global short-cut. The SWCNN has better performance than the CNN. One of the weaknesses of the SWCNN has low fault tolerance. If the the neuron is failed, the SWCNN shows lower fault tolerance than the CNN. Previously, we proposed TMR and Reliability Counter (RC) for fault tolerance the SWCNN. In this paper, we propose the Stateful Reliability Counter (Stateful RC) method to improve tolerance. The Stateful RC has a failure state of the last histrory. The Stateful RC for TMR has higher fault tolerant than TMR and RC in the low repaire rate.

Keywords: Fault Tolerance, Cellular Neural Networks, Small-World Networks, Small-World Cellular Neural Networks
Communicated by: D. Taniar & T. Sheltami

1 Introduction

Cellular Neural Networks (CNN) are used in pattern recognition, image processing and various other applications with changing templates. Small-World Cellular Neural Networks (SWCNN) are CNNs that incorporate small-world networks, which fall between regular and random networks, and can easily be applied to image processing [1, 2]. The SWCNN has feature lower fault tolerant than CNN. Previously, we proposed fault-tolerant SWCNNs for stop faults and intermitting faults. In this paper, we propose a Stateful Reliability Counter for use in the SWCNN for intermitting faults.

The outline of this paper is as follows. In Section 2 we define the SWCNN, while the proposed method is explained in Section 3 and 4. Section 5 describes the simulations and evaluations. Finally, we present our conclusions.

2 Small-World Cellular Neural Networks

In this section, we first describe Cellular Neural Networks and small-world networks. Thereafter, we define Small-World Cellular Neural Networks.

2.1 Cellular Neural Networks

A Cellular Neural Network (CNN) is a network in which the neuron elements are regularly arranged in an array, and connected in a lattice [2], [3]. A CNN can be applied to various applications, such as image processing and pattern recognition. The state (1) and output (2) equations are given below.

$$\dot{x}_{ij}(t) = -x_{ij}(t) + I + \sum_{c(kl)\in N_r(ij)} A(ij;kl)y_{kl}(t) + \sum_{c(kl)\in N_r(ij)} B(ij;kl)u_{kl}(t) \tag{1}$$

$$y_{ij}(t) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \tag{2}$$

Here, each element takes an input u_{ij}. The state variable x_{ij} is set by the input u_{ij} and output y_{ij}. The range of the output values is $-1 \leq y_{ij} \leq +1$. Templates explain which $c(ij)$ are connected to the neighborhood, where templates A and B are output and input templates, respectively. A is a feedback template dependent on output value y, while B is a feedforward template dependent on input value u. $N_r(ij)$ is a group of $c(ij)$'s neighboring cells at a distance r or less (Fig. 1). I is the bias.

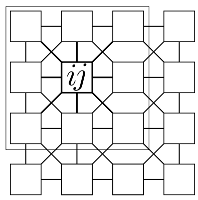


Fig. 1. CNN

2.2 Small-World Networks

An example of a small-world network is the global inter-connectedness of people, where any person is connected to any other in the world through a chain of 6 people on average. Watts and Strogatz have formulated small-world networks in terms of graph models [3, 4] by introducing two parameters:

- L: Average path length
- C: Clustering coefficient

L is the number of edges in the shortest path between two nodes, averaged over all pairs. C_v for node v has k links, with $k_v(k_v-1)/2$ edges between them. C is the average of all C_v. A feature of small-world networks is that C is large, while L is small (Fig. 2).

Small-world networks are, therefore, highly clustered with a short average path length. This network feature has rapidly become the trend. Fig. 2 illustrates a β-model which generates small-world networks that are regular networks rewired randomly with probability β. As β increases, the network becomes more random. Thus the small-world network falls between regular and random networks.

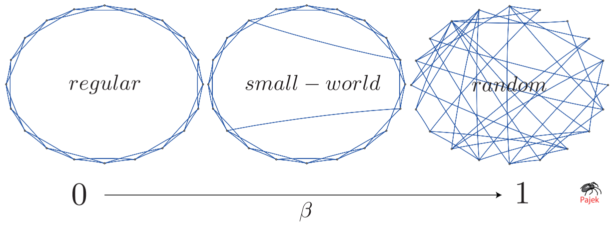


Fig. 2. Small-World Networks

2.3 Small-World Cellular Neural Networks

The Small-World Cellular Neural Network was proposed by Tsuruta et al. [2], and is a CNN with a small-world network structure. A SWCNN has at most one additional link to the CNN (Fig. 3) and has a faster convergence than the CNN [1, 2]. The state (3) is as follows:

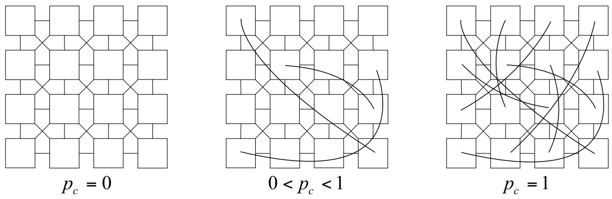


Fig. 3. SWCNN

$$\dot{x}_{ij}(t) = -x_{ij}(t) + I + \sum_{c(kl)\in N_{r(ij)}} A(ij;kl)y_{kl}(t) + \sum_{c(kl)\in N_{r(ij)}} B(ij;kl)u_{kl}(t) + w_c M(ij\ pq)y_{ij}(t) \tag{3}$$

Here, w_c is the connection weight for additional links and $M(ij;pq)$ denotes the connection coefficient between $c(ij)$ and $c(pq)$. When $c(ij)$ and $c(pq)$ are connected $M(ij;pq) = 1$; otherwise, $M(ij;pq) = 0$. The templates for noise reduction and edge detection in the SWCNN are given below. These templates are used for noise reduction in binary images.

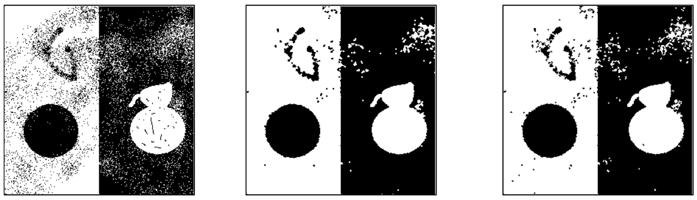
$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 0, w_c = 1.0 \tag{4}$$

Fig. 4(a) shows the original image, while Figs. 4(b) and (c) show the processed images. Figs. 4(b) and 4(c) have random connection rates, $r_c = 0$ and $r_c = 1$, respectively. In this example, the image processed by the SWCNN is better than that processed by the CNN.

Next, we use the following templates for edge detection in grayscale images.

Fig. 5(a) shows the original image, while Figs. 5(b) and (c) show the processed images with a random connection rate, $r_c = 0$ and $r_c = 1$, respectively. Once again, the image processed by the SWCNN is better than that processed by the CNN, showing that the SWCNN can detect edges more accurately than the CNN.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, I = 0, w_c = 1.0 \tag{5}$$



(a) Original image (b) CNN ($p_c = 0$) (c) SWCNN ($p_c = 1$)

Fig. 4. Noise Reduction

(a) Original image (b) CNN ($p_c = 0$) (c) SWCNN ($p_c = 1$)

Fig. 5. Edge Detection

3 Fault Tolerant methods for the SWCNN

In this section, we first describe the proposed fault-tolerant SWCNN. Two different methods are currently available, namely the TMR and the Reliability Counter TMR.

3.1 SWCNN using TMR

Triple Modular Redundancy (TMR) [5, 6] selects one of the outputs according to the votes from three modules executing the same three operations in parallel. TMR gracefully tolerates a great majority of faults.

The TMR method for a SWCNN [8], illustrated in Fig. 6, stacks three similar SWCNNs. Voting is performed by the three neurons located in the same vertical position and this determines the output. However, when three result are , it is not concidered.

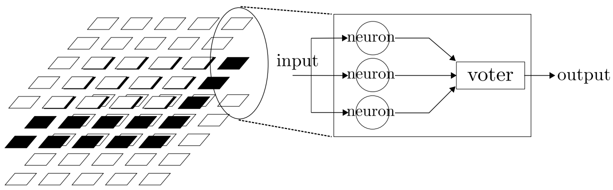


Fig. 6. TMR

3.2 SWCNN using Reliability Counter for TMR

We now describe the Reliability Counter (RC) for TMR[9], as illustrated in Fig. 7. There are 3 RCs, when each RC keeps the reliability of the corresponding neurons. The RC for each plane of neurons counts the number of times there is an inconsistent neuron in the voting procedure and in which that neuron is disregarded. The RC gives the expected level of reliability where the lower the level, the more reliable it is; 0 denotes the highest reliability. Basically, the output from three cells, processing the same operation simultaneously, is decided by majority voting using TMR. Also, if all 3 neuron output values are different, the output value is chosen from the neuron with the lowest RC value. The reliability level for each cell is decided as follows. When voting is performed, a cell in minority side, RC of the cell lowers reliability.

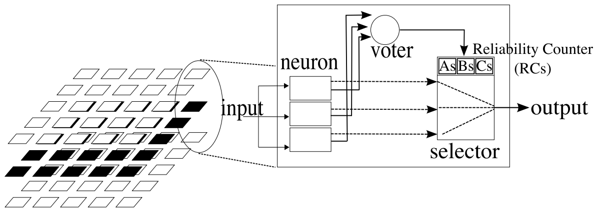


Fig. 7. Reliability Counter for TMR

Like the TMR, our proposed SWCNN with RC does not have sufficient fault tolerance for intermittent failures. In the next section, the reliability evaluation of an additional method with failure history is described.

4 Stateful Reliability Counter for TMR

Intermitting faults are faults where modules recover from a failure state to a normal state at a particular rate. In most case, a module fails in a failure state as a result of environmental effects, and consequently, the module is unable to handle intermitting faults itself. We introduce a method that considers the failure history of each module. In this section, we introduce the Stateful Reliability Counter for TMR. The RC method decides reliability using TMR, but has a low fault tolerance for intermitting faults because it does not consider these faults.

Fig. 8 illustrates the proposed Stateful Reliability Counter (Stateful RC) for TMR. Each Stateful RC depicts a particular neurons state. When the decision approved by majority voting is 2:1, the value of the counter is set to 1 or 0.

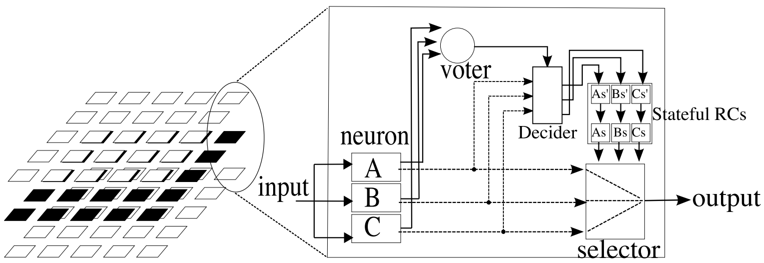


Fig. 8. Stateful RC

Here, we describe how a Stateful RC is set for state evaluation and the output of the neurons. The state of each neuron (failure or normal), is represented by a Current Stateful RC and a Next Stateful RC.

If the result of the voting by the neurons is unanimous, the output is set according to the unanimous value. Regarding the content of all three Current Stateful RCs, if the number of failure states is 0 or 1, all three Next Stateful RCs are set to the normal state. If there are two failures, the states of the Next Stateful RCs are set according to the states of the respective Current Stateful RCs. Regarding the content of one or more Current Stateful RCs that are in the minority according to the voting, if the respective Current Stateful RC is in a failure state, all Next Stateful RCs corresponding to the Current Stateful RCs on the majority side are set to the normal state. In addition, the output is selected according to the majority value. Regarding the content of a Current Stateful RC in the minority according to the voting, if the Current Stateful RC is in the normal state, all Next Stateful RCs corresponding to RCs on the majority side are set to the failure state and the Next Stateful RC corresponding to an RC on the minority side is set to the normal state. In addition, the output is selected according to the minority value.

The value of each Current Stateful RC is replaced by that of the corresponding Next Stateful RC at the next state decision.

5 Simulations and Evaluations

Section 5.1 describes the simulations and the evaluation method, while Section 5.2 gives the simulation results.

5.1 Simulations and Evaluation Method

The method for the simulations is summarized below.

- Fault Type: intermitting failure
  - Repair rate: 0.1 to 1.0
  - Failure rate: 0 to 0.1
- Tasks of cells: noise reduction and edge detection

In the case of an intermitting fault, a node fails temporarily. This means that the failed node is repaired at random. The difficulty with which it is repaired depends on the repair rate, which is the probability of repairing the failed cell autonomously in a clock cycle. For example, if the repair rate is 0, a failed neuron is never repaired and this is equivalent to a termination failure. If the repair rate is 1, once a node fails, it is repaired within the clock cycle. We evaluate the performance of the fault models using a matched rate, which is the ratio of the number of pixels that are different in the features of the reference image and those of the resulting simulation image to the total number of pixels. The matched rate is represented by Eq. (6).

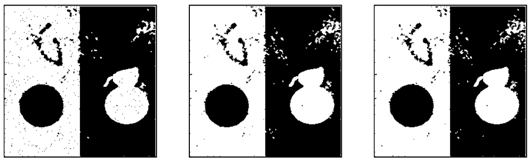
$$matched\ pixels\ rate = \frac{matched\ pixels}{all\ pixels} \tag{6}$$

5.2 Evaluation

Figs. 9 to 11 show the noise reduction simulation results, while Figs. 12 to 14 show the edge detection simulation results. Fig. 9 shows the simulation results for noise reduction when the repair rate is 1. In this simulation, TMR and RC produce less noise than the single SWCNN.

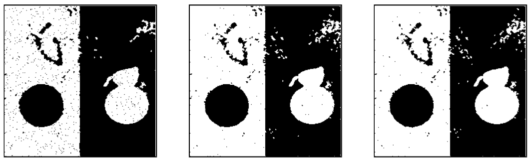
Fig. 10 shows the simulation results for noise reduction when the repair rate is 0.5. In this simulation, TMR and RC produce less noise than the single SWCNN. Fig. 11 shows the simulation results for noise reduction with a repair rate of 0.1. In this simulation, TMR and RC produce less noise than the single SWCNN. TMR and RC give similar results.

Fig. 12 shows the simulation results for edge detection with a repair rate of 1. In this simulation, TMR and RC produce less noise than the single SWCNN. Fig. 13 shows the



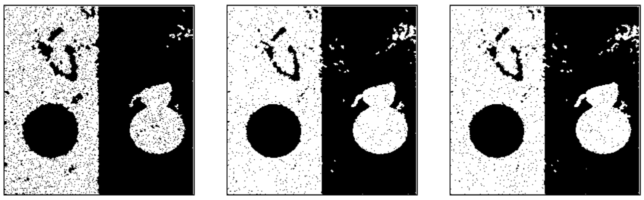
(a) SWCNN (b) TMR (c) RC

Fig. 9. Noise Reduction (failure rate = 0.01, repair rate = 1)



(a) SWCNN (b) TMR (c) RC

Fig. 10. Noise Reduction (failure rate = 0.01, repair rate = 0.5)



(a) SWCNN (b) TMR (c) RC

Fig. 11. Noise Reduction (failure rate = 0.01, repair rate = 0)

(a) SWCNN (b) TMR (c) RC

Fig. 12. Edge Detection (failure rate = 0.01, repair rate = 1)

simulation results for edge detection with a repair rate of 0.5. In this simulation, noise is included in the output images for both the SWCNN. Fig. 14 shows the simulation results

(a) SWCNN (b) TMR (c) RC

Fig. 13. Edge Detection (failure rate = 0.01, repair rate = 0.5)

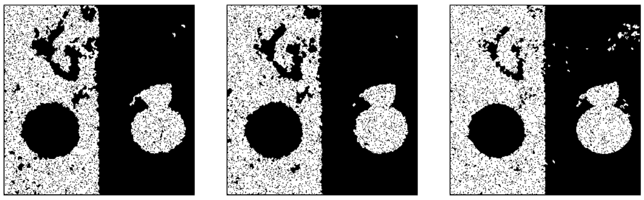
for edge detection when the repair rate is 0.1. In this simulation, the TMR and RC produce less noise than the single SWCNN. All two proposed methods exhibit greater fault tolerance than the SWCNN. There is however, not much difference between the results of the TMR and RC. In this example, the TMR and RC produce less noise than the single SWCNN. The two

(a) SWCNN (b) TMR (c) RC

Fig. 14. Edge Detection (failure rate = 0.01, repair rate = 0.1)

proposed methods are more fault tolerant than the SWCNN, with the TMR and RC giving almost identical results.

Next we evaluate the Stateful RC. Fig. 15 shows the noise reduction results for the various methods. TMR and RC give almost identical results, while the Stateful RC shows better



(a) TMR (b) RC (c) Stateful RC

Fig. 15. Repair rate: 0.1, Failure rate: 0.03

fault tolerance than both the TMR and RC. Fig. 16 shows the simulation results for edge

(a) TMR (b) RC (c) Stateful RC

Fig. 16. Repair rate: 0.1, Failure rate: 0.03

detection. TMR and RC give almost identical results, while the Stateful RC exhibits better fault tolerance than both the TMR and RC. Next, we show that the Stateful RC is effective for a wide range of repair rates (Figs. 17 to 22). Fig. 17 shows the noise reduction results for a repair rate of 0.1. In this simulation, the Stateful RC exhibits better fault tolerance than the TMR, RC, and SWCNN, while the TMR and RC show better fault tolerance than the SWCNN. The Stateful RC has the best performance at a repair rate of 0.1. Fig. 18 shows the edge detection results for the repair rate 0.1. In this simulation, the Stateful RC exhibits the best fault tolerance of all our methods. The TMR and RC have a lower fault tolerance than the SWCNN for high failure rates. For the repair rate 0.1, the Stateful RC is the most fault tolerant. Fig. 19 shows the noise reduction results for a repair rate of 0.5. In this simulation, the TMR and RC are more fault tolerant than the Stateful RC, which is, in turn, more fault tolerant than the SWCNN. Fig. 20 shows the edge detection results for the repair rate 0.5.

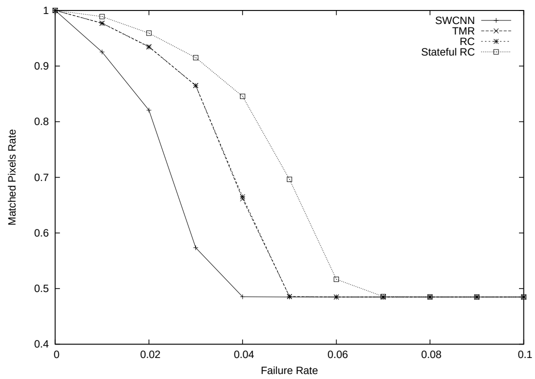


Fig. 17. Noise Reduction (repair rate: 0.1)

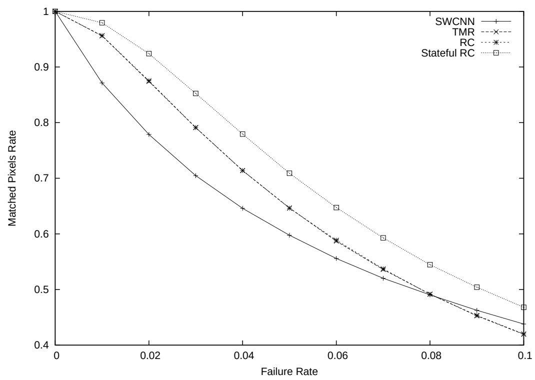


Fig. 18. Edge Detection (repair rate: 0.1)

In this simulation, the TMR and RC are more fault tolerant than the Stateful RC, which is, in turn, more fault tolerant than the SWCNN. At a repair rate of 0.5, the Stateful RC is less fault tolerant than both the TMR and RC methods.

Fig. 21 shows the noise reduction results for the repair rate 1. In this simulation, the TMR and RC exhibit greater fault tolerance than the Stateful RC, which, in turn, exhibits greater fault tolerance than the SWCNN. Fig. 22 shows the edge detection results for the repair rate 1. In this simulation, the TMR and RC have greater fault tolerance than the Stateful RC, which in turn, has greater fault tolerance than the SWCNN. Overall at a repair rate of 1, the

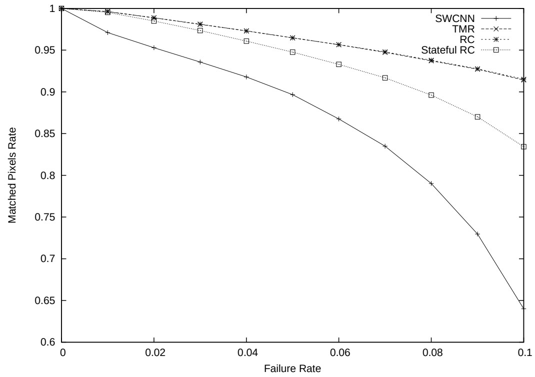


Fig. 19. Noise Reduction (repair rate: 0.5)

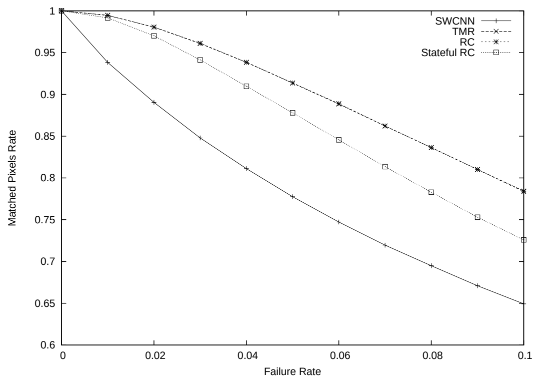


Fig. 20. Edge Detection (repair rate: 0.5)

Stateful RC is not good. The Stateful RC exhibits less fault tolerance than both the TMR and RC at this repair rate, as well as at a repair rate of 0.5.

Although the Stateful RC has good fault tolerance at low repair rates, it does not have good fault tolerance at high repair rates.

6 Conclusions

In the event of a neuron failure, the SWCNN shows lower fault tolerance than the CNN. In this paper, we proposed a fault tolerant SWCNN using majority voting including a reliability

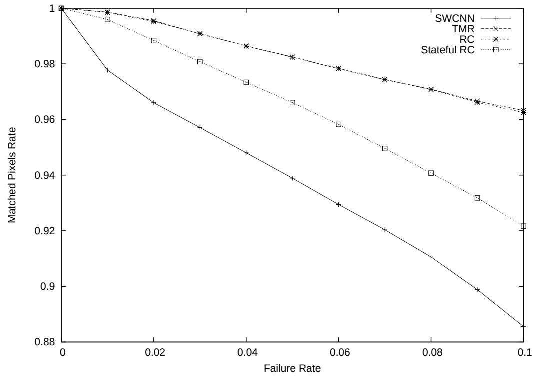


Fig. 21. Noise Reduction (repair rate: 1)

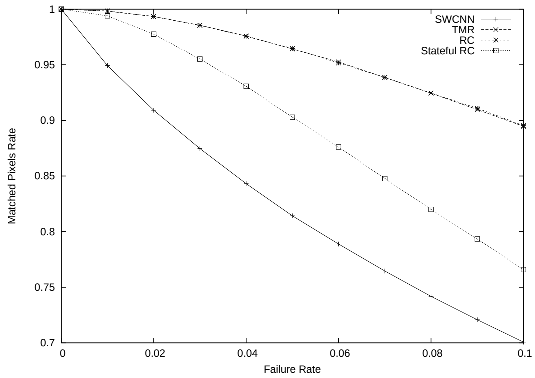


Fig. 22. Edge Detection (repair rate: 1)

evaluation technique. In addition, the reliability evaluation incorporates a failure state history assessment of the presence of fault occurrences (Stateful RC). Fault tolerance has been simulated using the repair rate as a parameter. When the repair rate is low, the Stateful RC shows higher fault tolerance, whereas when the repair rate is high, both the RC and TMR show higher fault tolerance.

In conclusion, the effectiveness of the RC has been confirmed to be good for SWCNN applications, while the Stateful function works better with a low repair rate. As future work, a further study of the method of acquiring state information is important.

Reference

1. L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257-1272, 1988.
2. K. Tsuruta, Z. Yang, Y. Nishio, and A. Ushida, "Small-world cellular neural networks for image processing applications," *European Conference on Circuit Theory and Design*, vol. 1, pp. 225-228, 2003.
3. D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440-442, 1998.
4. M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167-256, 2003.
5. D. K. Pradhan, *Fault-tolerant computing system design*. Prentice Hall PTR, 1996.
6. M. Abd-El-Barr, *Design and Analysis of Reliable and Fault-Tolerant Computer System*. Imperial College Press, 2007
7. K. Matsumoto, M. Uehara, and H. Mori, "Fault tolerant small-world cellular neural networks," in *Proc. 4th Int. Symp. Frontiers in Networking with Applications (FINA2008)*, 2008, pp. 168-172.
8. K. Matsumoto, H. Mori, and M. Uehara, "Fault tolerance in small-world cellular neural networks for image processing," in *Proc. 21st Int. Conf. Advanced Information Networking and Applications*, vol. 1, 2007, pp. 835-839.
9. K. Matsumoto, H. Mori, and M. Uehara, "Fault tolerance for small-world cellular neural networks," in *Network-Based Information Systems (NBIS2008), Springer LNCS*, vol. 5186, 2008, pp.223-231.