AIS1/PRJ1-4L
(Late-News Paper)

Image Recognition Using Oxide Semiconductor Crossbar Memristors with Implementation of Slit Detection and Local Autonomous Learning

Yuta Takishita[1], Mutsumi Kimura[1,2], Yasuhiko Nakashima[1]

Takishita.yuta.tt6@is.naist.jp
[1]Nara Institute of Science and technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
[2]Ryukokou University, 1-5 Yokotani, Seta Oe-cho, Otsu, Shiga 520-2194, Japan
Keywords: crossbar-memristor, neural-network, neuromorphic, local-autonomous-learning

ABSTRACT

In this study, we simulated an image recognition system using visual cortex layers and xbar memristor as a full connection layer. The simulation results show that the best accuracy was 69% when the MNIST datasets were used, and it was 23% when the CIFAR-10 datasets were used.

1 INTRODUCTION

In the existing Neumann computers [1], the memory unit and the arithmetic unit are completely separated, and the performance has been improved by the development of the semiconductor miniaturization technology. However, because of the end of Moore's law [2], it is considered that there will be no further improvement in semiconductor miniaturization technology in the future, and the performance improvement of the Neumann computers will slow down. According to recent artificial intelligence (AI) [3] research, it has been found that AI calculation works if it has a calculation accuracy of about 8 bits[4]. Therefore, elements using various materials have been devised every year. For example, HfOx is used in [5] and memristive magnetic tunnel junction are used in [6].

A model called a neural network that mimics the neural circuit of the brain is often used for AI, and recently, research on neuromorphic computers that implement the neural network at the hardware level has been conducted. It is considered that the neuromorphic computers can operate at a low power consumption with significantly higher calculation efficiency than the conventional Neumann computers. Neuromorphic computers include those that use digital elements such as CMOS [7]–[9] and those that use analog elements [10][11].

A convolutional neural network (CNN) [12] is a type of neural network mainly used for image recognition, and its prototype can be found in Neocognitron [13], which was devised based on the neurophysiological knowledge of the visual cortex (V1) of the brains of organisms. Neocognitron is a neural network in which convolutional layers corresponding to simple cells for feature extraction and pooling layers corresponding to complex cells having a function of allowing positional displacement are alternately arranged in a hierarchical manner. In Neocognition, learning was done by self-organization, but in CNN, learning was done by backpropagation. However, learning by backpropagation requires a huge amount of calculation. In the human primary visual cortex, edge detection, contour coordination, and slit detection are possible [14].

In this study, we fixed the weight of the kernel in the feature extraction layer, eliminated the need for kernel learning, and aimed at a structure that can be implemented in hardware. In addition, we have developed a learning algorithm that utilizes the electrical characteristics of oxide semiconductors for the full connection (FC) layer. Then, the behavior was simulated by using the developed algorithm for actual image recognition. The detailed information on the prior precondition behind this study was also published elsewhere [15]–[17]. In particular, in this paper, we will show the simulation results of the best accuracy when the MNIST datasets and the CIFAR-10 datasets are used. Moreover, we will show the dependency of the training rate and deviation of initial resistance value of the oxide semiconductor.

2 EXPERIMENT

2.1 Architecture

Normally, a CNN is used for image recognition, and it has the structure shown in Fig.1. However, because the purpose of this research is to implement whole system as a hardware, image recognition was performed by replacing each layer of the CNN with the proposed method as shown in Fig.2. Some letter and image patterns are inputted to these architectures.

2.2 Visual Cortex

Usually, the CNN based on the structure shown in Fig.1 is often used for image recognition. Here, it is necessary to train the kernel weights of the convolutional layer for extracting the feature quantity. However, training kernel weights requires a large amount of calculation and is expensive to optimize. Therefore, by creating layers that mimic the structure of the primary visual cortex, as shown in Fig.2, we reduced the learning cost and made it possible to implement them in hardware. In these layers, edge detection and slit detection are performed. In the edge detection, the edges were

detected by comparing the Sum of Absolute Difference (SAD) value in four directions with the threshold in a 3x3 area. In the slit detection, a kernel as shown in Fig.3 was used so that the feature of the line angle could be detected at every 22.5 degree. Fig.4 show the detected degrees using the slit detector.

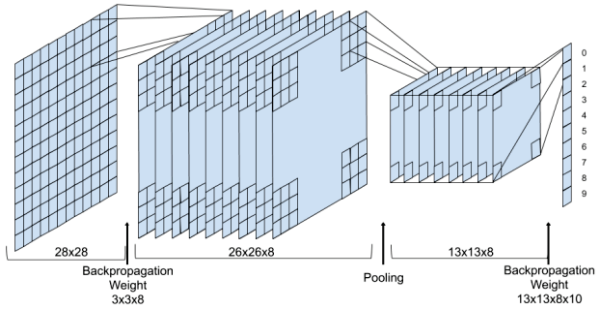

Fig. 1 Conventional CNN architecture

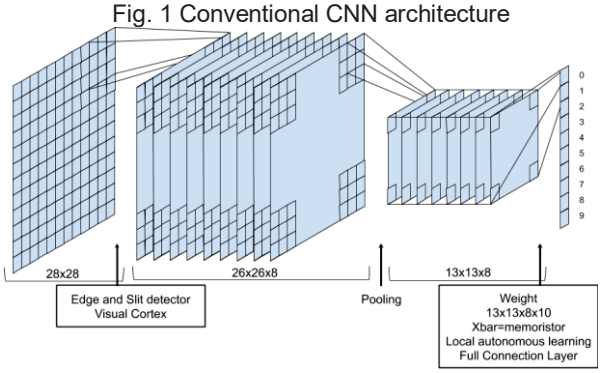

Fig. 2 Proposed architecture with implementation of the slit detector and local autonomous learning

Fig. 3 Implemented slit type for every 22.5 degree

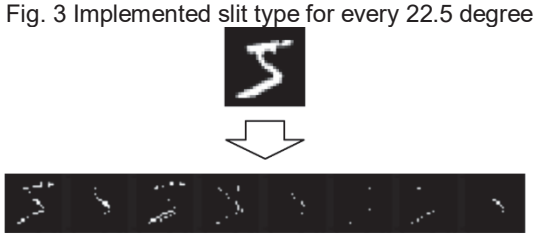

Fig. 4 Detected degree

2.3 Full Connection Layer

Neural networks are often used also for the FC layers of ordinary CNNs, and backpropagation is often used for training. However, since the oxide semiconductor memristor is used as the FC layer in this study, backpropagation cannot be used because extremely complicated circuits or systems are needed to control the electrical characteristics. Therefore, we have developed a training algorithm that does not use backpropagation. The oxide semiconductor memristor is characterized in that its resistance value changes according to the number of times the voltage is applied [18], and the structure is shown in Fig.5. Here, a thin film of an oxide semiconductor is sandwiched between the top electrode and the bottom electrode. The voltage is applied between the top electrode and the bottom electrode, and the electric current flows through the oxide semiconductor. Furthermore, that the characteristics is shown in Fig.6. Here, it is modeled that the electric current gradually increases by applying the voltage. In addition, the structure is a simple xbar like that shown in Fig.7, so high integration is possible. In our research, we assumed to use a memristor using Ga-Sn-O. Here, the top and bottom electrodes are patterned to multiple buslines, and the oxide semiconductor memristors are formed at the cross points of these buslines.

In the training, the resistance value at the designated place is changed by inputting the training data corresponding the image patterns from one electrode and label data from the opposite electrode, as shown in Fig.5 and Fig.7. Here, the training rate is defined as a rate of the decrease of the resistance per voltage application for the initial resistance. In the simulation, the deviation of the initial resistance value of the oxide semiconductor was changed from 0% to 40%, the learning rate was changed from 10^{-1} to 10^{-7}, and the influence on the recognition accuracy was investigated. Moreover, we investigated the effect when the number of the electrodes for inputting label data changes.

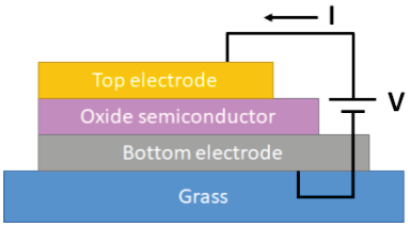

Fig. 5 Cross sectional structure of the oxide semiconductor memristors

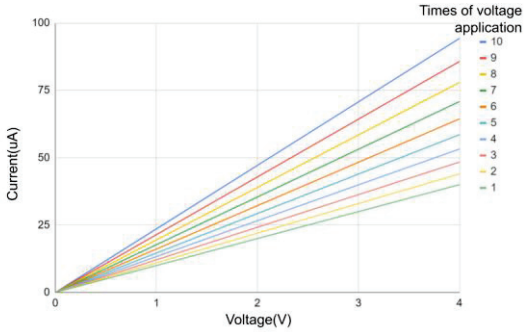

Fig. 6 Simulation model of memoristor characteristics

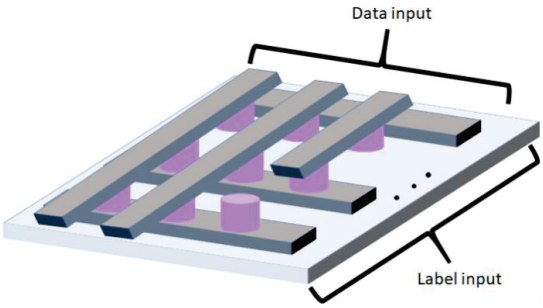

Fig. 7 Xbar shape for the oxide semiconductor memristors

3 RESULTS and DISCUSSION

3.1 MNIST datasets

Dependence of the recognition accuracy on the training rate and resistance deviation for MNIST datasets is shown in Fig.8. It was found that the larger the deviation in resistance value, the lower the recognition accuracy. When the recognition accuracy was maximized, the deviation in the initial resistance value was 0%, the training rate was 10^{-7}, and the recognition accuracy was 69%. When the deviation of the initial resistance value was 0%, the recognition rate was high for the training rate from 10^{-7} to 10^{-4}, but for the training rate above 10^{-3} the recognition accuracy decreased. It is considered that the reason is that if the training rate is too large, the amount of change in the resistance value of the frequently learned part is too large and the influence of other parts is ignored. Moreover, when the resistance variation is from 10% to 40%, the recognition accuracy was maximized at the training rate of 10^{-3} or 10^{-4}. It is considered that the reason is that if the training rate is too small, the deviation in the initial resistance value cannot be overcome by training.

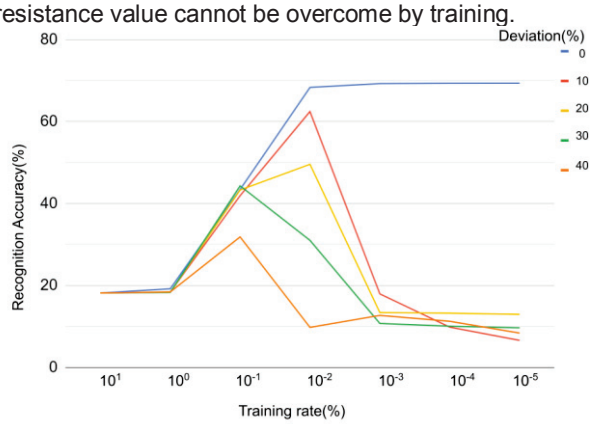

Fig. 8 Dependence of the recognition accuracy on the training rate and resistance deviation for MNIST datasets

3.2 CIFAR-10 datasets

Dependence of the recognition accuracy on the training rate and resistance deviation for CIFAR-10 datasets is shown in Fig.9. Similar to the dependence for MNIST, it was found that the larger the deviation in resistance value, the lower the recognition accuracy. When the recognition accuracy became the maximum, the deviation of the initial resistance value was 0%, the training rate was 10^{-7}, and the recognition accuracy was 23%. When the deviation of the initial resistance value was 0%, the recognition rate was high for the training rate from 10^{-7} to 10^{-3}, but for the training rate above 10^{-2} the recognition accuracy decreased. It is considered that the reason is that if the training rate is too large, the amount of change in the resistance value of the frequently learned part is too large and the influence of other parts is ignored. Moreover, when the resistance variation is from 10% to 40%, the recognition accuracy was maximized at the training rate of 10^{-3}. It is considered that the reason is that if the training rate is too small, the deviation in the initial resistance value cannot be overcome by training. These discussions for CIFAR-10 are roughly the same as those for MINST. In addition, the training rate when the recognition accuracy is maximum in Fig.8 and Fig.9 has different values because the image size, data size, etc. are different between MINST and CIFAR-10 datasets. From these results, it was found that it is necessary to find a suitable training rate for each data set.

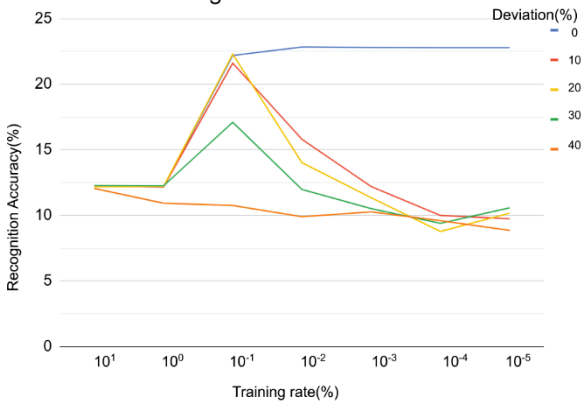

Fig. 9 Dependence of the recognition accuracy on the training rate and resistance deviation for CIFAR-10 datasets

3.3 Label Input Method

Incidentally, the abovementioned results were obtained by applying a label input to a bar of the xbar shape. In addition to the abovementioned results, the other results were also obtained by applying a label input to some other bars for MINIST datasets as shown in Fig.10. From these results, it was found that the recognition accuracy for "a bar and one randomly-selected bar" and "a bar and two randomly-selected bars" is the same as that for a bar. The reason may be that randomly-selected bars do not give bias during training. Moreover, it was also found that the recognition accuracy for "a bar and one neighboring bar" and "a bar and two neighboring bars" has less recognition accuracy. It is considered that the reason is that training is biased because correct and incorrect labels are used at the same time for each learning.

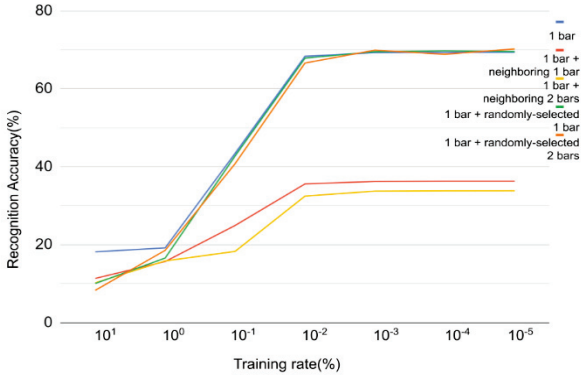

Fig. 10 Accuracy of MNIST dataset with changing the number of trained bar

4 CONCLUSIONS

We investigated image recognition using xbar memristor with implementation of slit detection and local autonomous learning. In this image recognition program, the edge detection layer and the slit detection layer, which have the same function as the primary visual cortex for feature extraction and do not require training, are used. Moreover, we developed a training algorithm when using a device made of an oxide semiconductor memristor in the shape of xbar for AI training. The developed training algorithm was implemented in an image recognition program to simulate the behavior. In the simulation, the deviation in the initial resistance value of the oxide semiconductor and the training rate were changed. The recognition accuracy was up to 69% when using the MNIST datasets and up to 23% when using the CIFAR-10 datasets. In addition, it was found that the training rate when the recognition accuracy was maximized was different when using different datasets, so it is necessary to use the training rate suitable for each dataset.

5 ACKNOWLEDGMENTS

This work is partially supported by KAKENHI (C) 19K11876.

REFERENCES

[1] J. Von Neumann and M. D. Godfrey, “First Draft of a Report on the EDVAC,” IEEE Ann. Hist. Comput., Vol. 15, No. 4, pp. 27–75, (1993).

[2] F. Peper, “The End of Moore’s Law: Opportunities for Natural Computing?,” New Gener. Comput., Vol. 35, No. 3, pp. 253–269, (2017).

[3] McCarthy, Minsky, Rochester, and Shannon, “1 Automatic Computers 2 . How Can a Computer be Programmed to Use a Language 4 . Theory of the Size of a Calculation,” pp. 1–13, (1955).

[4] N. Wang, J. Choi, D. Brand, C. Y. Chen, and K. Gopalakrishnan, “Training deep neural networks with 8-bit floating point numbers,” Adv. Neural Inf. Process. Syst., Vol. 2018-Decem, No. NeurIPS, pp. 7675–7684, (2018).

[5] B. Gao, Y. Bi, H. Chen, R. Liu, P. Huang, B. Chen, L. Liu, X. Liu, S. Yu, H. S. P. Wong, and J. Kang, “Ultra-low-energy three-dimensional oxide-based electronic synapses for implementation of robust high-accuracy neuromorphic computation systems,” ACS Nano, Vol. 8, No. 7, pp. 6998–7004, (2014).

[6] P. Krzysteczko, J. Münchenberger, M. Schäfers, G. Reiss, and A. Thomas, “The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system,” Adv. Mater., Vol. 24, No. 6, pp. 762–766, (2012).

[7] A. Neckar, S. Fok, B. V, Benjamin, T. C. Stewart, N. N. Oza, A. R. Voelker, C. Eliasmith, R. Manohar, and K. Boahen, “Braindrop: A Mixed-Signal Neuromorphic Architecture with a Dynamical Systems-Based Programming Model,” Proc. IEEE, Vol. 107, No. 1, pp. 144–164, (2019).

[8] J. Hsu, “IBM’s new brain [News],” IEEE Spectr., Vol. 51, No. 10, pp. 17–19, (2014).

[9] D. Ma, J. Shen, Z. Gu, M. Zhang, X. Zhu, X. Xu, Q. Xu, Y. Shen, and G. Pan, “Darwin: A neuromorphic hardware co-processor based on spiking neural networks,” J. Syst. Archit., Vol. 77, pp. 43–51, (2017).

[10] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, and J. P. Strachan, “Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine,” Adv. Mater., Vol. 30, No. 9, pp. 1–10, (2018).

[11] X. Zhang, A. Huang, Q. Hu, Z. Xiao, and P. K. Chu, “Neuromorphic Computing with Memristor Crossbar,” Phys. Status Solidi Appl. Mater. Sci., Vol. 215, No. 13, pp. 1–16, (2018).

[12] L. Haoxiang and Lin, “A convolutional neural network approach for face identification,” Proc. CVPR, pp. 5325–5334, (2015).

[13] K. Fukushima and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” Pattern Recognit., Vol. 15, No. 6, pp. 455–469, (1982).

[14] K. Toyama, “Structual and Functional Design of Visual System,” IPSJ Magazine, Vol. 26, No. 2. pp. 108–116, (1985).

[15] Y. Takishita, M. Kobayashi, K. Hattori, T. Matsuda, S. Sugisaki, Y. Nakashima, and M. Kimura, “Memristor property of an amorphous Sn-Ga-O thin-film device deposited using mist chemical-vapor-deposition method,” AIP Adv., Vol. 10, No. 3, (2020).

[16] M. Kimura, Y. Koga, H. Nakanishi, T. Matsuda, T. Kameda, and Y. Nakashima, “In-Ga-Zn-O thin-film devices as synapse elements in a neural network,” IEEE J. Electron Devices Soc., Vol. 6, No. 1, pp. 100–105, (2017).

[17] M. Kimura, K. Umeda, K. Ikushima, T. Hori, R. Tanaka, J. Shimura, A. Kondo, T. Tsuno, S. Sugisaki, A. Kurasaki, K. Hashimoto, T. Matsuda, T. Kameda, and Y. Nakashima “Neuromorphic system with crosspoint-type amorphous Ga-Sn-O thin-film devices as self-plastic synapse elements,” ECS Transactions, 90:157-166, 04 (2019).

[18] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, “Nanoscale memristor device as synapse in neuromorphic systems,” Nano Lett., Vol. 10, No. 4, pp. 1297–1301, (2010).