

# Synaptic Device Network Architecture with Feature Extraction for Unsupervised Image Classification

Sungho Kim, Bongsik Choi, Meehyun Lim, Yeamin Kim, Hee-Dong Kim, and Sung-Jin Choi\*

For the efficient recognition and classification of numerous images, neuro-inspired deep learning algorithms have demonstrated their substantial performance. Nevertheless, current deep learning algorithms that are performed on von Neumann machines face significant limitations due to their inherent inefficient energy consumption. Thus, alternative approaches (i.e., neuromorphic systems) are expected to provide more energy-efficient computing units. However, the implementation of the neuromorphic system is still challenging due to the uncertain impacts of synaptic device specifications on system performance. Moreover, only few studies are reported how to implement feature extraction algorithms on the neuromorphic system. Here, a synaptic device network architecture with a feature extraction algorithm inspired by the convolutional neural network is demonstrated. Its pattern recognition efficacy is validated using a device-to-system level simulation. The network can classify handwritten digits at up to a 90% recognition rate despite using fewer synaptic devices than the architecture without feature extraction.

## 1. Introduction

Recently, deep learning algorithms have made substantial advances. They have surpassed other machine learning algorithms, especially in the fields of image recognition, speech recognition, and translation.<sup>[1]</sup> In particular, the convolutional neural network (CNN) architecture that consists of multiple convolution and subsampling (pooling) layers can extract the useful feature information without requiring significant manual input data engineering,<sup>[2]</sup> which leads to higher accuracy in various

pattern recognition applications. However, the emulation of deep learning algorithms in conventional digital computing systems requires a significant energy consumption<sup>[3]</sup> due to the intrinsic drawback of current digital computing systems (i.e., the limited data transfer rate between the memory and the central processing unit (referred to as the von Neumann bottleneck)). Therefore, highly energy-efficient computing system architectures are one of the key aspects in the emerging computing paradigm, along with the development of software algorithms (such as machine learning). The capabilities of future computing systems should involve recognizing, performing computations, and responding in real time to big data.

Neuromorphic systems are expected to break through the von Neumann bottleneck, which is deriving inspiration from

the biological neural system.<sup>[4]</sup> Interestingly, the neuromorphic system has the memory and the processing units coexisting at the same physical location. More precisely, the processing and storing of information can be simultaneously performed by modulating the connection strength of the synaptic device (i.e., synaptic weight) following the appropriate learning rules (e.g., spike-timing-dependent plasticity (STDP)<sup>[5]</sup>). Since the synaptic device is the most abundant and a key functional element of the neuromorphic system, considerable research efforts have been made to implement an artificial synaptic device by exploiting emerging analog-type resistive switching devices based on two-terminal (typically known as memristors)<sup>[6–11]</sup> or three-terminal<sup>[12–14]</sup> structures. Furthermore, primitive levels of functional neuromorphic systems have been experimentally demonstrated for the applications of pattern classification,<sup>[15]</sup> analog-to-digital conversion,<sup>[16]</sup> principal component analysis,<sup>[17]</sup> sparse coding calculations,<sup>[18]</sup> and unsupervised learning system.<sup>[19,20]</sup>

However, the implementation of a functional neuromorphic system is still very challenging, particularly with the understanding of how the characteristics of the single synaptic device affect the performance of the entire system. Since the performance of the system depends on the architecture, the learning rules as well as the synaptic device functionalities, a quantitative device-to-system level analysis that takes into account all of these is indispensable. Nevertheless, most previous studies have been limited to the single device-level analysis.<sup>[6–14]</sup>

Prof. S. Kim, Prof. H.-D. Kim  
Department of Electrical Engineering  
Sejong University  
Seoul 05006, South Korea

B. Choi, Y. Kim, Prof. S.-J. Choi  
School of Electrical Engineering  
Kookmin University  
Seoul 02707, South Korea  
E-mail: sjchoiee@kookmin.ac.kr

Dr. M. Lim  
Mechatronics R&D Center  
Samsung Electronics  
Gyeonggi-do 18448, South Korea

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/sml.201800521>.

DOI: 10.1002/sml.201800521

Therefore, the demonstration of the synaptic device has not yet directly led to the continuous optimization of the neuromorphic system. In particular, in the case of pattern recognition applications, only a few studies have been reported that employ the highly efficient feature extraction algorithm of the CNN on the hardware neural network.<sup>[21,22]</sup> This increases the performance gap between the machine learning algorithms (software) and the neuromorphic system (hardware).

This study addresses these issues by utilizing a device-to-system level simulation based on a developed learning rule that has the potential for unsupervised online learning and consequential image classification in a synaptic device network. It is expected that our results can be used to help with the quantitative design and optimization of synaptic devices, especially the required number of weight states and the variation margin of the weight update to improve the pattern recognition accuracy. In addition, we redesign a synaptic device network architecture including a feature extraction process inspired by the CNN, and improved image classification efficacy is validated by the simulation. The proposed network can classify the Modified National Institute of Standards and Technology database (MNIST) handwritten digits at up to a 90% recognition rate despite using fewer synaptic devices than the architecture without feature extraction.

## 2. Results and Discussion

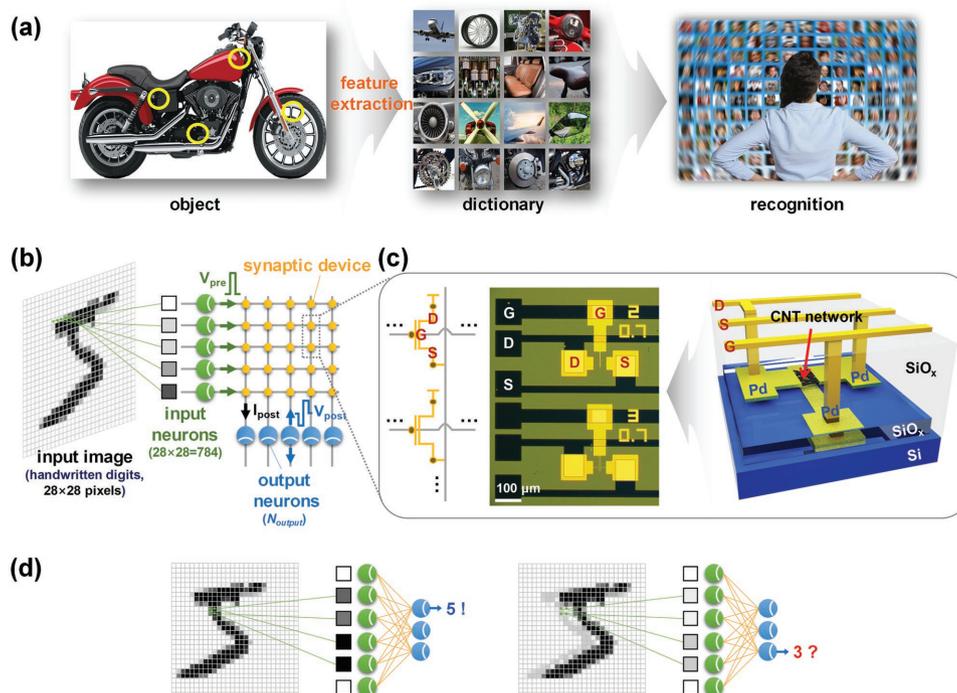
Feature extraction is a key principle of the image recognition processes in human and animal vision systems. A feature can be roughly defined as an “interesting” part of an object. Efficiently distinguishing or recognizing an object becomes possible not by judging the entire image pixel-by-pixel but rather by comparing only the extracted features and learned memories with each other, as shown in **Figure 1a**. Theoretically, feature extraction reduces the dimensionality of the original data into a new space based on identified features, and it has been applied in many computer vision and machine learning algorithms.<sup>[23,24]</sup> Among the machine learning algorithms, the CNN architecture is inspired by the organization of the human’s visual cortex, and the architecture is comprised of a number of convolutional and subsampling (pooling) layers.<sup>[1,2]</sup> These particular layers result in the efficient extraction of invariant features from the input patterns, which enables one to overcome the limit of previous deep belief networks. Consequently, the CNN has showed overwhelming performance in image and speech recognition applications.

In our previous work,<sup>[25,26]</sup> we experimentally demonstrated an artificial synaptic device to emulate the functionalities of biological synapses and designed the synaptic device network along with a learning rule for the pattern recognition. Briefly, **Figure 1b** shows the implemented synaptic device network and the learning rule for our conceived image classification system. With the crossbar layout, each input neuron is connected with one pixel of the image. Input neurons emit presynaptic spikes ( $V_{pre}$ ) wherein the timing of the presynaptic spikes represents the analog information of the pixel intensities. Subsequently, presynaptic spikes from the input neurons can simultaneously trigger multiple synaptic transistors. Postsynaptic currents

( $I_{post}$ ) determined by the channel conductance of each synaptic transistor are collected and accumulated at an output neuron. If the accumulated postsynaptic current level is greater than a given threshold value, one output neuron fires a postsynaptic spike ( $V_{post}$ ). Then, the synaptic weight can be modulated to any analog state according to the correlation of the pre and postsynaptic spikes. **Figure 1c** shows the demonstrated synaptic device network (array) of our carbon nanotube (CNT) transistor based synaptic devices (see the Experimental Section).<sup>[25,26]</sup>

According to the aforementioned learning rule (which was referred to “simplified STDP scheme” in our previous work), the network could classify the MNIST handwritten digits with 70% recognition accuracy.<sup>[25,26]</sup> However, our previous study still had two limitations. 1) It is unclear what the required specification of the synaptic device is in order to obtain higher recognition accuracy. Although the analog modulation of the weight state at the synaptic device is the key factor of the learning rule, a quantitative analysis on how the number of weight states ( $N_{state}$ ) and the variation margin of the weight update ( $\Delta G = G_{max}/G_{min}$ ) affect the recognition accuracy has not been performed in our previous studies. 2) Additionally, the network and the learning rule could not take into account the spatial structure of the images. For example, although the input images (digit “5”) are similarly distant on exactly the same footing as shown in **Figure 1d**, the network and the learning rule treat these two cases as totally different images, which is contradictory to the way humans deal with visual images. To train such spatially different images, more training images, and a bigger network are required. It is a critical drawback that restricts the energy-efficient and highly accurate recognition system. Fortunately, the same problem has been considered in the deep belief network algorithm, and the CNN architecture has solved this issue by adapting the particular feature extraction algorithm.<sup>[1]</sup> Therefore, in this study, 1) we investigate how different weight modulation characteristics of the synaptic device affect the recognition accuracy in the network by using a device-to-system level simulation. 2) In addition, inspired by the CNN, the feature extraction algorithm (software) is demonstrated by the synaptic device network (hardware), and we investigate the improvement of the recognition accuracy in the redesigned network.

First, we investigate the impact of the synaptic device’s specifications on the recognition accuracy in the network. (Note that the following discussion is independent of the synaptic device structure or weight modulation mechanism.) The analog modulation of the weight at the synaptic device (i.e., conductance,  $G$ ) can be characterized by several aspects: here we focus on  $N_{state}$  and  $\Delta G$ . When a pulse train is applied to our synaptic device for the potentiation (increasing  $G$ ) or the depression (decreasing  $G$ ), the measured  $G$  shows a gradual transition (**Figure 2a**) in which  $N_{state}$  and  $\Delta G$  can be manipulated by adjusting the number of applied pulses and the level of the pulse ( $V_{LTP}$ ,  $V_{LTD}$ ), respectively. Here, since the experimental demonstration of a reliable intermediate conductance of more than eight states is still very challenging in the case of two-terminal synaptic devices,<sup>[27–29]</sup> the analysis of the required  $N_{state}$  value is essential to provide the design guidelines for the synaptic device network. To identify the effect of  $N_{state}$  on the recognition accuracy, the device-to-system level simulation was carried out.<sup>[25,26]</sup> The



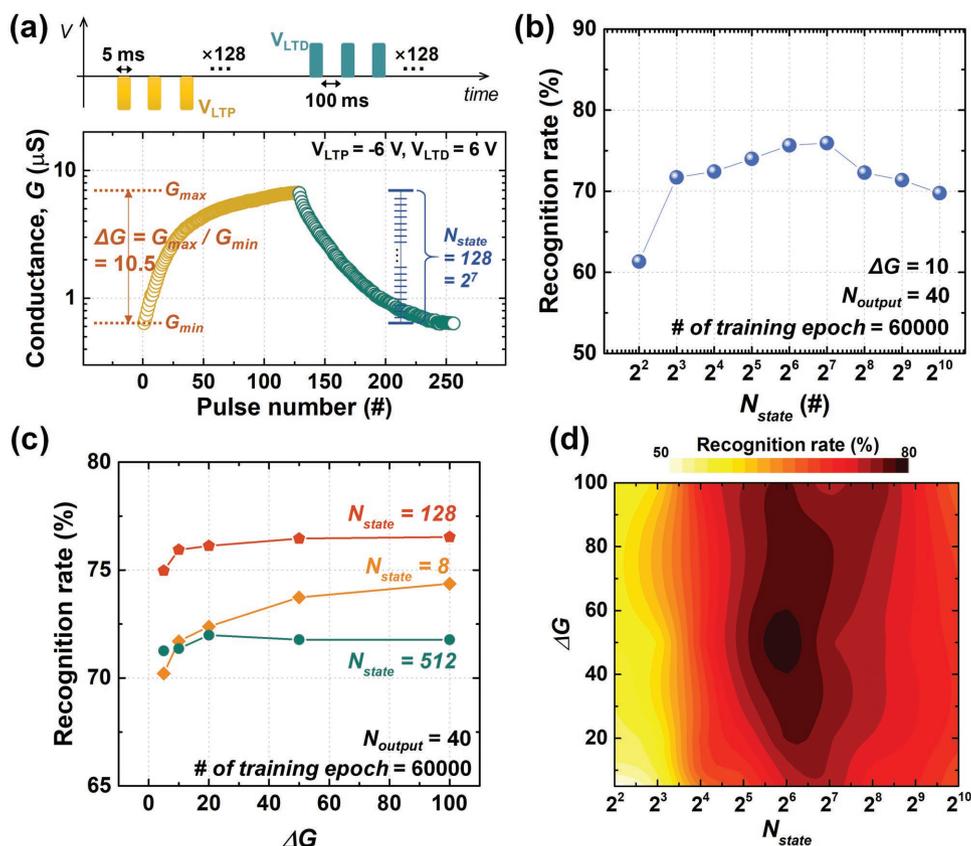
**Figure 1.** a) The schematic of the human visual process with feature extraction. The feature extraction is any algorithm that transforms raw data into features that can be used as an input for a learning algorithm. b) The synaptic device network for pattern recognition of  $28 \times 28$  grayscale images consisting of the input and output layers. The input neuron is fully connected to the input image pixel in a one-to-one manner. The synaptic devices are located at the junctions between the input and output neurons. c) The fabricated synaptic device network and the schematic of the synaptic device. The CNT-based synaptic transistor emulates the functions of biological synapses through the analog channel's conductance modulation. d) The example of the spatial difference problem that has similarly occurred in the deep learning algorithm.

detailed simulation procedure, parameters, and model used in this study are described in Note S1 of the Supporting Information. Note that the effect of nonlinearity in conductance modulation is neglected in the simulation to simplify the analysis. In other words, it is assumed in the simulation that the amount of modulation in conductance is constantly changed as shown in Figure S1 (Supporting Information). Figure 2b shows the recognition rate (i.e., classification accuracy) for the test images as a function of  $N_{\text{state}}$ . Obviously, the recognition rate can be improved by increasing  $N_{\text{state}}$ . When  $N_{\text{state}} = 128$ , the maximum recognition rate reaches 75%. However, an  $N_{\text{state}}$  greater than 128 rather leads to the degradation of the recognition rate. This is because a larger  $N_{\text{state}}$  requires more training epochs to modulate the weight up to the desired level. In other words, more pulse trains are needed to obtain the desired conductance change of the synaptic device, which results in the inefficiency (slowdown) of training process (see Note S2 of the Supporting Information). Moreover, compared to the case of  $N_{\text{state}} = 8$ , the recognition rate is improved by only 5% at  $N_{\text{state}} = 128$ . Therefore, the effect of  $N_{\text{state}}$  is not critical for improving the recognition rate.

Similar analysis was performed regarding the effect of  $\Delta G$  on the recognition accuracy. In our synaptic device,  $\Delta G$  is controllable by adjusting the level of the applied pulse ( $V_{\text{LTP}}$ ,  $V_{\text{LTD}}$ ) (see Note S3 of the Supporting Information). Figure 2c shows the simulated recognition rate as a function of  $\Delta G$ . Interestingly, the recognition rate can be evidently improved by increasing  $\Delta G$  when  $N_{\text{state}}$  is small (e.g.,  $N_{\text{state}} = 8$ ). In contrast, the recognition

rate is rarely improved in spite of increasing  $\Delta G$  when  $N_{\text{state}}$  is large (e.g.,  $N_{\text{state}} = 128$  or 512). These results indicate that the effects of  $\Delta G$  and  $N_{\text{state}}$  on the recognition rate are complicatedly conjugated. Although the recognition rate can be improved by increasing  $\Delta G$  or  $N_{\text{state}}$ , the improvement is limited by the combined effect of  $\Delta G$  with  $N_{\text{state}}$ . Consequently, increasing  $\Delta G$  or  $N_{\text{state}}$  are not always advantageous for improving the recognition rate, but the specific optimum values only can improve the recognition rate, as shown in Figure 2d. However, note that the improved recognition rate by increasing  $\Delta G$  is still only 5%, even in the best case. Therefore, it is obvious that the tuning of synaptic device specification ( $\Delta G$  and  $N_{\text{state}}$ ) is not effective for improving the recognition rate. It implies that further research efforts should be devoted to better develop the architecture and learning rule.

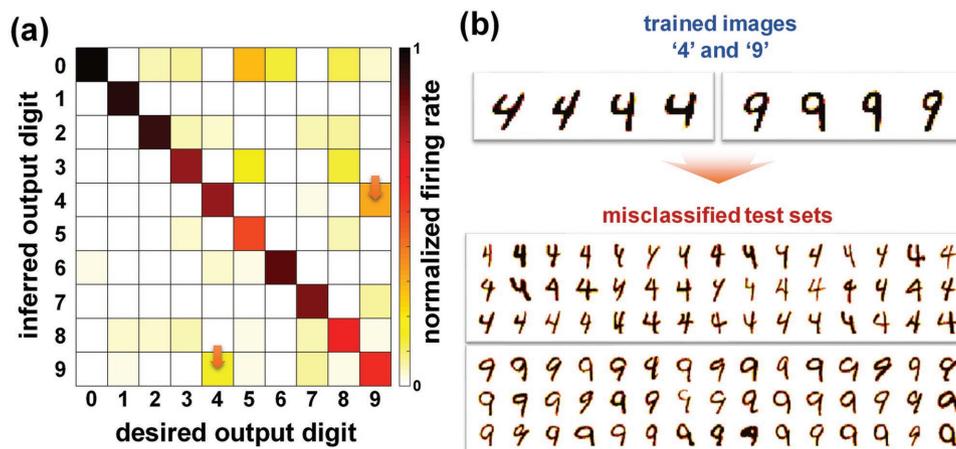
Next, to understand the nature of misclassification, we investigate the details of the misclassified images. Figure 3a shows the confusion matrix over ten digits of the MNIST test set. In this, every single classification of the test inputs belongs to one of the  $10 \times 10$  tiles, and its position is determined by the actual digit and inferred digit. Given a recognition rate of  $\approx 70\%$ , the most typical mistakes were that "4" was misidentified as "9" and "9" was misidentified as "4." Additionally, Figure 3b shows the trained images (i.e., the maps of synaptic weights associated with each output neuron) in the network and the misclassified test sets in the case of "4" and "9" digits. Surprisingly, the images that are misclassified are all easily distinguishable with our own eyes. Nonetheless, the reason why these images are



**Figure 2.** a) Schematics of the applied pulse trains used to measure the analog conductance modulation of our synaptic device. Each pulse train consists of 128 potentiation or depression pulses applied to the gate ( $V_{LTP}$  and  $V_{LTD}$  for 5 ms), followed by small, nonperturbative read voltage pulses (1 V for 100 ms) within the intervals. b) The simulated recognition rate as a function of  $N_{state}$  after 60 000 times of training epochs ( $\Delta G$  is fixed to 10, and  $N_{output}$  is fixed to 40). c) The simulated recognition rate as a function of  $\Delta G$  and  $N_{state}$ . d) The summarized correlation of  $\Delta G$  and  $N_{state}$  that impacts on the recognition rate.

misclassified is due to a spatial difference. Although misclassified images look almost identical in appearance to the trained images, they shifted by a very small difference. Unfortunately, our learning rule considers these differences to be completely

different images. In fact, this issue can be easily solved by just increasing the number of output neurons ( $N_{output}$ ). When  $N_{output}$  increases, since the number of trained images corresponding to each output neuron increases, spatially different images can be



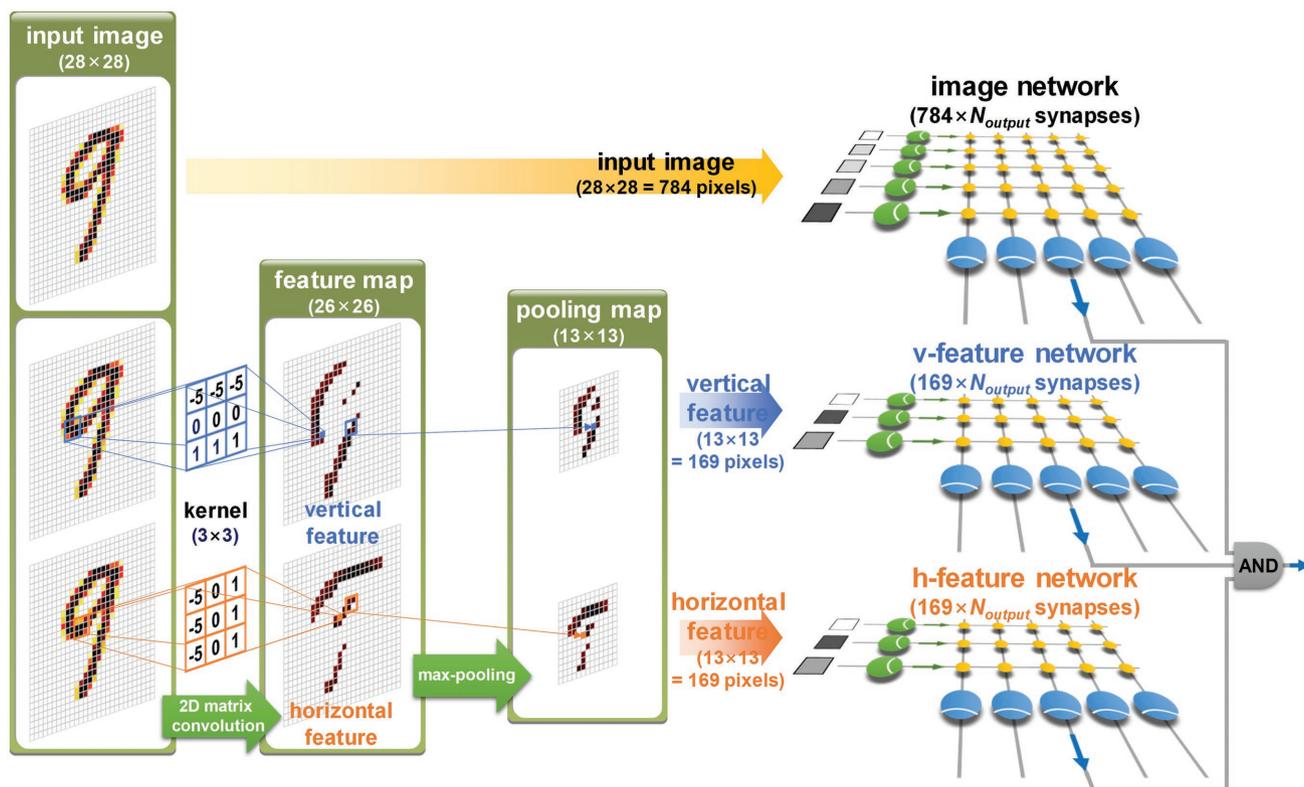
**Figure 3.** a) Average confusion matrix of the testing results over the ten digits of the MNIST test set. High values along the identity indicate a correct identification, whereas high values anywhere else indicate confusion between two digits, such as with the digits “4” and “9.” b) The selected examples of the trained images and the misclassified test sets in the case of the “4” and “9” digits.

distinguished through each output neuron. Consequently, the recognition rate can be improved up to 80% by only increasing  $N_{\text{output}}$  without any adjustments to the synaptic device's specifications (see Note S4 of the Supporting Information). However, this approach requires a bigger network with more synaptic devices and consequently more energy consumption, which is far from what the neuromorphic system seeks.

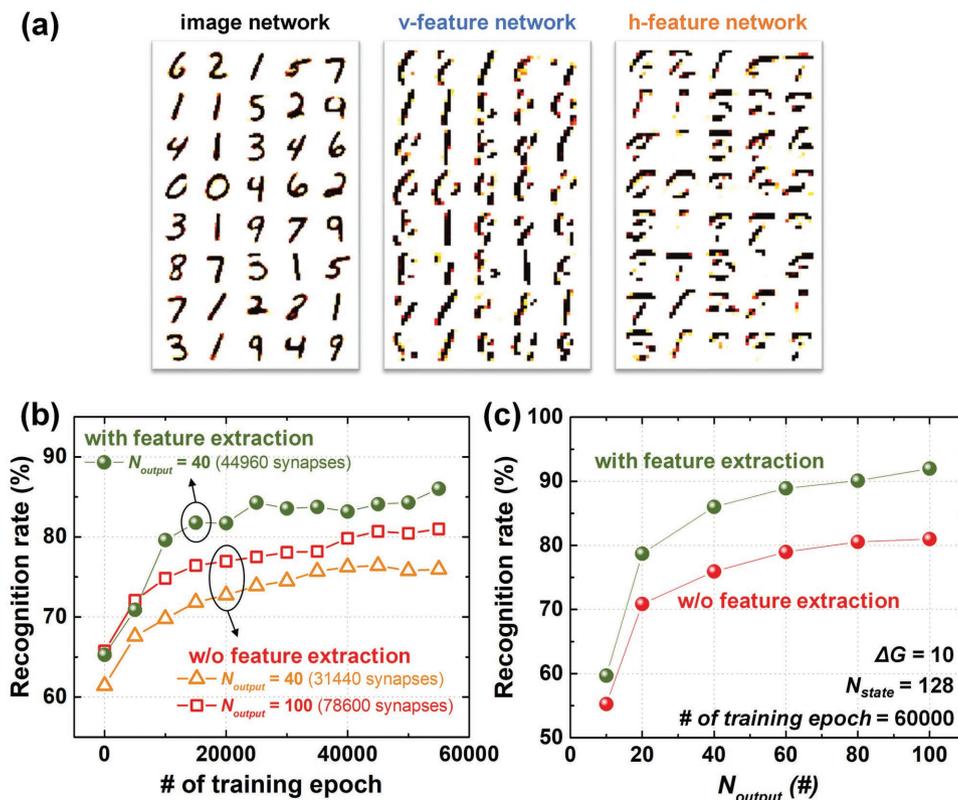
To overcome this issue, in the following, we redesigned the synaptic device network architecture by including the hardware-based feature extraction process. **Figure 4** shows the schematics of the proposed feature extraction architecture that is inspired by the CNN. The main network plays the role of training an original input image ( $28 \times 28 = 784$  pixels) through the same learning rule as mentioned above, which is named the "image network." The image network consists of a total of  $784 \times N_{\text{output}}$  synaptic devices. Additionally, we add two particular networks that are named the "v-feature network" and the "h-feature network," respectively. These two networks play the role of training the extracted vertical/horizontal features obtained by performing the convolution and subsampling (pooling) processes. In detail, the convolution operation is individually performed on each pixel of the input image, where the convolution kernel is a 2D matrix (e.g., a  $3 \times 3$  matrix). Each pixel value of the input image is multiplied by the corresponding value in the kernel, and the consequential sum of products becomes the pixel value in the feature map. Here, since the edge is one of important features of the image, we use a well-defined simple edge detector kernel called the Prewitt kernel.<sup>[30]</sup> Two different types of

Prewitt kernels can extract the vertical and the horizontal edges, respectively. Since the input image has  $28 \times 28$  pixels and a  $3 \times 3$  matrix kernel, the feature map will be  $26 \times 26$  pixels. This is because the kernel can be moved 26 pixels across or down before colliding with the right-hand side or bottom of the input image. Next, this feature map is subjected to a subsampling (pooling) process that condenses the spatial information of the feature map. For instance, each pixel value in the pooling map summarizes a region of  $2 \times 2$  pixels in the feature map. When we use the common procedure of max-pooling, the pooling map outputs only the maximum value of the feature map in the  $2 \times 2$  region. Therefore, the pooling map will be reduced to  $13 \times 13$  pixels. Since this pooling process discards the exact positional information, the extracted features are less sensitive to spatial differences, which is the same strategy used in the CNN. This pooling map is trained in the v-feature and h-feature networks. Consequently, the v-feature and the h-feature networks consist of a total of  $169 \times N_{\text{output}}$  synaptic devices, respectively.

**Figure 5a** shows all trained images in the "image network," the "v-feature network," and the "h-feature network," respectively (e.g.,  $N_{\text{output}} = 40$ ), where the vertical and horizontal edges of the images are clearly trained by the v-feature and h-feature networks, respectively. Under this situation, when the test image to be classified is input to these three networks, the original test image is compared to the trained image in the image network. Then, the firing occurs at the specific output neuron corresponding to the best matched the trained image. Likewise, the extracted features of the test image through the convolution and



**Figure 4.** The schematic of the proposed feature extraction architecture that is inspired by the CNN. The image network trains an original input image ( $28 \times 28 = 784$  pixels). In addition, the v-feature network and h-feature networks ( $13 \times 13 = 169$  pixels) train the extracted vertical/horizontal features obtained by performing the convolution and subsampling (pooling) processes.



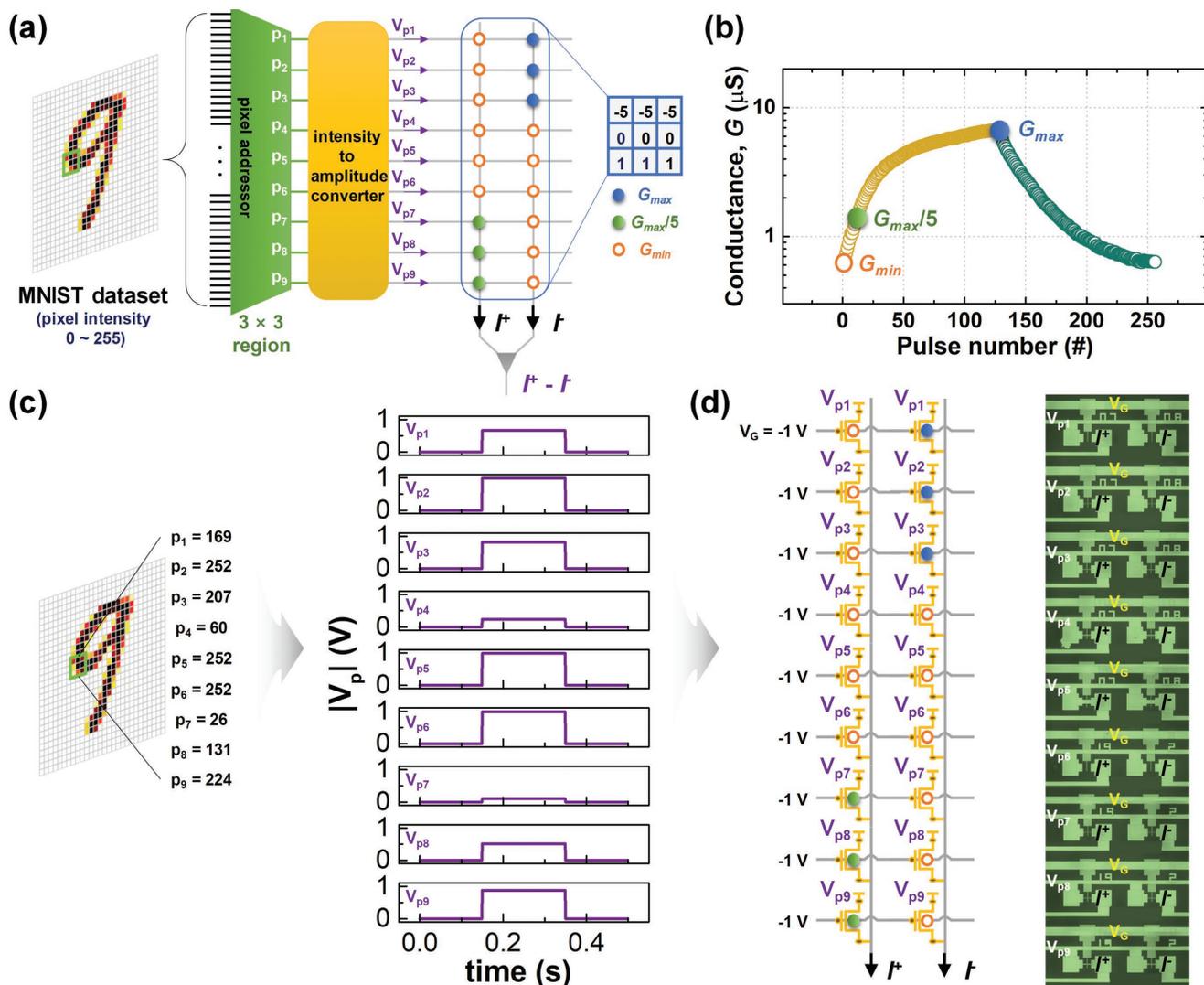
**Figure 5.** a) The synaptic weights in the “image network,” the “v-feature network” and the “h-feature network,” respectively ( $N_{output} = 40$ ). b) The simulated recognition rate with or without the feature extraction architecture. The feature extraction architecture demonstrates a higher recognition rate despite using fewer synaptic devices than the architecture without feature extraction c) The simulated recognition rate as a function of  $N_{output}$  with or without feature extraction architecture.

pooling processes are compared to the trained features in the v-feature and h-feature networks, respectively, and firings occur at the specific output neurons that are best matched with the trained features. If the test image leads to a firing on the same output neurons in all three networks, it means that the test image matches both the shape of the trained image and the vertical/horizontal features. Therefore, the possibility that the image is properly classified increases. Figure 5b shows the simulated recognition rate according to the number of training epochs. Note that the proposed feature extraction architecture demonstrates a higher recognition rate despite using fewer synaptic devices than the previous architecture without feature extraction. Obviously, the feature extraction can effectively improve the recognition rate while reducing the energy consumption owed to the use of fewer synaptic devices. Furthermore, by increasing the number of output neurons ( $N_{output}$ ) (as shown in Figure 5c), it becomes possible to implement a hardware-based image classification system with a recognition rate exceeding 90%.

The last part in this study is how to implement the convolution process in the hardware. In the following, we present the experimental demonstration of the convolution operation by using the simple crossbar synaptic device network,<sup>[21]</sup> as shown in Figure 6a. (At this time, the pooling process requires additional complex circuitry that could not be experimentally implemented as of yet.) In principle, the MNIST dataset consists of  $28 \times 28$  pixels in which each pixel corresponds to the intensity of

the image in the range of  $0 \approx 255$ . Among these pixels,  $3 \times 3$  pixels on which the convolution will be performed are selected, and the value of the intensity proportionally corresponds to the pulse amplitudes ( $0 \text{ V} \leq V_{p1} \text{ to } V_{p9} \leq 1 \text{ V}$ ). These pulses are applied to the crossbar network. They generate the current according to the channel conductance ( $G$ ) of each cross-point synaptic device in which the channel conductance is adjusted to three different values depending on the desired kernel weight values (e.g.,  $G_{max}$ ,  $G_{max}/5$ , and  $G_{min}$ , as shown in Figure 6b). Then, the total output column currents ( $I^+$  and  $I^-$ ) are the sum of the multiplication between  $V_p$  and  $G$  (i.e.,  $I^+$  and  $I^- = \sum V_p \cdot G$ ). Note that this crossbar network must have two columns due to the negative weight value of the kernel. One value represents the sum of the products from the positive weight values in the kernel ( $I^+$ ), and the other represents the sum of the products from the negative weight values in the kernel ( $I^-$ ). Accordingly, a differential readout of the two column currents ( $I^+ - I^-$ ) represents the convolution results.

For the experimental proof-of-the-concept, the convolution operation was performed on a digit “9” image from the MNIST dataset, as shown in Figure 6c (the same method as that of ref. [21] was used).  $V_{p1} \approx V_{p9}$  are determined to be proportional to each pixel value (intensity), which is applied to the drain electrodes of each synaptic device in the network (Figure 6d). As a result, the output column currents (i.e.,  $I^+$  and  $I^-$ ) are the sum of the drain currents from each column, and  $I^+ - I^-$  represents



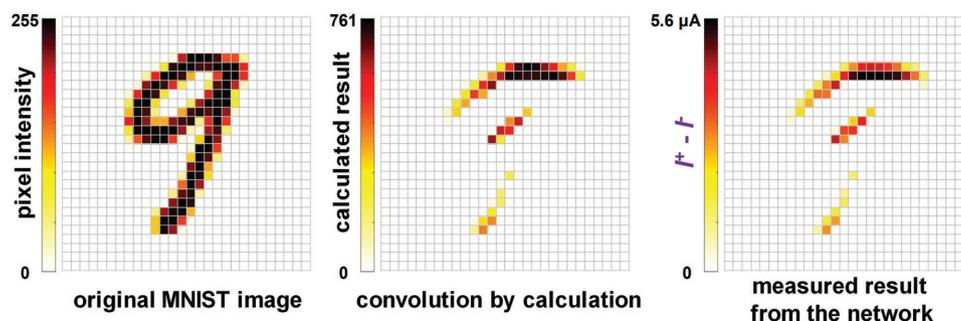
**Figure 6.** a) The process of the convolution operation using a simple crossbar synaptic device network. b) The selected channel conductance values ( $G_{max}$ ,  $G_{max}/5$ , and  $G_{min}$ ) that depend on the weight values of the kernel. c) A digit “9” image is input to the synaptic device network for the experimental convolution operation.  $V_{p1}$ – $V_{p9}$  are determined in proportion to each pixel value (intensity), and d) applied to the drain electrode of each synaptic device. The output column currents ( $I^+$  and  $I^-$ ) and consequential convolution result ( $I^+ - I^-$ ) represent the convolution result. The microscopic top-view image shows our fabricated  $9 \times 2$  crossbar network for the convolution operation where carbon nanotube-based synaptic transistors are located at the cross-point junctions of the rows and columns. Interestingly, since our synaptic device is based on the structure of the transistor (i.e., 3-terminal structure), the sneaky path current problem is perfectly prohibited.

the convolution results. **Figure 7** shows the convolution operation using the horizontal kernel (negative values after convolution operation are clipped to zero). The convolution result from the calculation is obviously consistent with the result from the measured data, which indicates that the convolution operation can be reliably performed using the simple synaptic device network. Therefore, the demonstrated methodology can be applied to implement the above-discussed efficient feature extraction architecture for the image classification system.

### 3. Conclusion

In summary, we have demonstrated a synaptic device network architecture with a feature extraction algorithm inspired

by the CNN, and the pattern recognition efficacy is validated using a device-to-system level simulation. Surprisingly, the proposed network can classify handwritten digit images at up to a 90% recognition rate using only a smaller number of synaptic devices than the architecture without feature extraction. This architecture has not been constructed in any other previous study. In addition, the simulation results can be used to help with the quantitative design and optimization of the synaptic devices. The tuning of synaptic device specification (i.e., the required  $N_{state}$  and  $\Delta G$ ) was not critical in improving the classification accuracy. Furthermore, we have experimentally demonstrated the convolution operation from the Prewitt vertical/horizontal kernels for the edge feature extraction on the fabricated  $9 \times 2$  crossbar synaptic device network. The demonstrated methodology is an important step toward effective big data



**Figure 7.** (left) The intensity of example image (digit “9”). (center) The calculated ideal convolution result. (right) Measured Prewitt horizontal kernel operation.

manipulation through the analog hardware implementation for more complex neuromorphic systems.

## 4. Experimental Section

**Fabrication of Carbon Nanotube-Based Synaptic Transistor Array:** CNT synaptic transistors were initially fabricated on highly p-doped rigid silicon substrates with a thermally grown 50 nm thick SiO<sub>2</sub> layer. We used the local back-gate structure for efficient local modulation of the channels in the CNT transistors. To form the local back-gate, the palladium (Pd) layer was first deposited and subsequently patterned using evaporation and a lift-off process, respectively. Next, a 50 nm thick SiO<sub>x</sub> layer, a 10 nm thick Au layer, and a 20 nm thick SiO<sub>x</sub> layer were sequentially deposited. The thin Au layer served as a floating gate for charge storage. Then, the top surface of the SiO<sub>x</sub> layer was functionalized with a 0.1 g mL<sup>-1</sup> poly-L-lysine solution to form an amine terminated layer that acted as an effective adhesion layer for the deposit of the CNTs. Subsequently, the CNT network channel was formed by immersing the chip into a 0.01 mg mL<sup>-1</sup> 99% semiconducting CNT solution (NanoIntegris, Inc.) for several hours, followed by a thorough rinse with isopropanol and deionized water. Subsequently, the source/drain electrodes consisting of Ti and Pd layers (each 2 and 40 nm, respectively) were deposited and patterned using conventional thermal evaporation and a lift-off process, respectively. Finally, additional photolithography and oxygen plasma steps were conducted to remove unwanted electrical paths, which isolated the devices from one another.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

S.K. and B.C. contributed equally to this work. This research was supported by the Nano-Material Technology Development Program (Grant No. NRF-2016M3A7B4910430) and the Basic Science Research Program (Grant Nos. NRF-2016R1D1A1B03930162, 2016R1A2B4011366, and 2016R1A5A1012966) through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning. This work was partially supported by the Future Semiconductor Device Technology Development Program (Grant No. 10067739) funded by MOTIE (Ministry of Trade, Industry & Energy) and KSRC (Korea Semiconductor Research Consortium).

## Conflict of Interest

The authors declare no conflict of interest.

## Keywords

carbon nanotubes, feature extraction, image classification, neuromorphic systems, recognition rates

Received: February 6, 2018

Revised: May 7, 2018

Published online:

- [1] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, 521, 436.
- [2] Y. Bengio, A. Courville, P. Vincent, *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE, **2013**, 35, 1798.
- [3] R. Andri, L. Cavigelli, D. Rossi, L. Benini, in *2016 IEEE Comput. Soc. Annu. Symp. VLSI*, Pittsburgh, USA July **2016**, pp. 236–241.
- [4] C. Mead, *Proc. IEEE* **1990**, 78, 1629.
- [5] G. Q. Bi, M. M. Poo, *J. Neurosci.* **1998**, 18, 10464.
- [6] C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Perez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, B. Linares-Barranco, *Front. Neurosci.* **2011**, 5, 26.
- [7] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, B. Linares-Barranco, *Front. Neurosci.* **2013**, 7, 2.
- [8] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, W. Lu, *Nano Lett.* **2010**, 10, 1297.
- [9] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, H. S. P. Wong, *IEEE Trans. Electron Devices* **2011**, 58, 2729.
- [10] S. Kim, C. Du, P. Sheridan, W. Ma, S. Choi, W. D. Lu, *Nano Lett.* **2015**, 15, 2203.
- [11] C. Du, W. Ma, T. Chang, P. Sheridan, W. D. Lu, *Adv. Funct. Mater.* **2015**, 25, 4290.
- [12] L. Q. Zhu, C. J. Wan, L. Q. Guo, Y. Shi, Q. Wan, *Nat. Commun.* **2014**, 5, 333.
- [13] J. Shi, S. D. Ha, Y. Zhou, F. Schoofs, S. Ramanathan, *Nat. Commun.* **2013**, 4, 84508.
- [14] F. Alibart, S. Pleutin, D. Guérin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat, D. Vuillaume, *Adv. Funct. Mater.* **2010**, 20, 330.
- [15] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, 521, 61.
- [16] X. Guo, F. Merrikh-Bayat, L. Gao, B. D. Hoskins, F. Alibart, B. Linares-Barranco, L. Theogarajan, C. Teuscher, D. B. Strukov, *Front. Neurosci.* **2015**, 9, 488.

- [17] S. Choi, J. H. Shin, J. Lee, P. Sheridan, W. D. Lu, *Nano Lett.* **2017**, *17*, 3113.
- [18] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W. D. Lu, *Nat. Nanotechnol.* **2017**, *12*, 784.
- [19] A. Serb, J. Bill, A. Khat, R. Berdan, R. Legenstein, T. Prodromakis, *Nat. Commun.* **2016**, *7*, 12611.
- [20] A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, E. Eleftheriou, *Nat. Commun.* **2017**, *8*, 1115.
- [21] L. Gao, P.-Y. Chen, S. Yu, *IEEE Electron Device Lett.* **2016**, *37*, 870.
- [22] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, C. Di Nolfo, P. Datta, A. Amir, B. Taba, M. D. Flickner, D. S. Modha, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 11441.
- [23] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York **2006**.
- [24] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Springer, Boston, MA **1993**.
- [25] S. Kim, J. Yoon, H. D. Kim, S. J. Choi, *ACS Appl. Mater. Interfaces* **2015**, *7*, 25479.
- [26] S. Kim, B. Choi, M. Lim, J. Yoon, J. Lee, H.-D. Kim, S.-J. Choi, *ACS Nano* **2017**, *11*, 2814.
- [27] A. Prakash, J. Park, J. Song, J. Woo, E.-J. Cha, H. Hwang, *IEEE Electron Device Lett.* **2015**, *36*, 32.
- [28] V. K. Nagareddy, M. D. Barnes, F. Zipoli, K. T. Lai, A. M. Alexeev, M. F. Craciun, C. D. Wright, *ACS Nano* **2017**, *11*, 3010.
- [29] W. Kim, A. Chattopadhyay, A. Siemon, E. Linn, R. Waser, V. Rana, *Sci. Rep.* **2016**, *6*, 36652.
- [30] I. E. Abdou, W. K. Pratt, *Proc. IEEE* **1979**, *67*, 753.