In-material reservoir implementation of reservoir-based convolution

Yuichiro Tanaka*, Yuki Usami*†, Hirofumi Tanaka*†, Hakaru Tamukoh*†
* *Research Center for Neuromorphic AI Hardware, Kyushu Institute of Technology*, Kitakyushu, Japan
† *Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology*, Kitakyushu, Japan

***Abstract*—This study aims to implement a reservoir-based convolutional neural network (CNN) on physical reservoir computing (RC) to develop an efficient image recognition system for edge AI. Therefore, we propose a novel reservoir-based convolution circuit system that uses in-material reservoir computing, a type of physical RC made from a sulfonated polyaniline network. The experimental results demonstrate that the proposed circuit system extracts image features in the same way as the original CNN and that a reservoir-based CNN on the in-material RC achieves an accuracy rate of 81.7% in an image classification task while an echo state network-based CNN achieves 87.7%.**

***Index Terms*—reservoir computing, reservoir-based convolution, in-material reservoir computing, edge AI**

I. Introduction

Most artificial intelligence (AI) technologies currently in use are based on deep learning (DL) [1] and achieve state-of-the-art results in many tasks such as image recognition tasks. A significant amount of training data and high-performance computers with graphics processing units, both of which are necessary to improve the DL performances and accelerate DL computations, have been used to support this DL success. Conversely, the requirements for edge AI are different from those for current AI: the amount of training data, computational resources, and especially consuming power are strictly limited. This study aims to realize an edge AI with low training costs that can be trained with little data and only a few computational costs, resulting in a low-power implementation.

The reservoir computing (RC) [2], [3] based approach is one of the solutions for the low-training cost AI realization. An RC is a type of recurrent neural network (NN) consisting of input, reservoir, and output layers. General NNs optimized using the backpropagation [4] update all the weight connections in networks, whereas the RC updates only a portion of weight connections. Therefore, the RC requires less training data and has lower computational costs than general NNs. Although the weight connections in the reservoir layer are fixed, RC can achieve comparable results to the general NNs in several tasks [5] because of its nonlinearity and high dimensionality.

RC can be used for image recognition tasks. An echo state network (ESN) [2], one of the RC implementations, achieved an accuracy rate of more than 99% on the MNIST dataset [6] [7]. An NN consisting of convolution layers and an ESN also achieved an accuracy of more than 99% and 86% on the MNIST and the Fashion-MNIST datasets [8], respectively [9], [10]. A deep delayed feedback reservoir achieved an accuracy of more than 60% on the CIFAR-10 dataset [11], [12]. A reservoir-based convolutional NN (CNN) that extracts image features with various spatial frequencies outperformed conventional RC-based networks in the image recognition tasks of the Fashion-MNIST and the CIFAR-10 datasets [13].

Several studies have implemented RCs on hardware to develop low-power RC systems, including Honda and Tamukoh's implementation of an ESN with 100 reservoir nodes on a field-programmable gate array (FPGA) [14], and Alomar et al. and Huang et al.'s implementation of ESNs with 300 nodes on FPGAs [15] [16]. Several studies have proposed hardware-oriented RC models for efficient hardware implementation. For example, Alomar et al. and Loomis et al. implemented stochastic computing-based RC with 50 nodes on FPGAs [17] [18], and Kawashima et al. implemented a chaotic Boltzmann machine (CBM) [19] based RC with 2,048 nodes on an FPGA [20].

Although these semiconductor-based hardware implementations are already more efficient than the current software implementations, this study aims to achieve an even more efficient implementation using physical RCs [21], which replace reservoir layers in RC with physical dynamical systems such as optical systems [22], spintronics devices [23], and soft materials [24]. Since physical RCs directly use physical phenomena as computations, their computations can be much more efficient than conventional systems.

As a first step toward the high-efficient RC implementation for image recognition, this study proposes a novel physical reservoir circuit system of the reservoir-based CNN using an in-material RC [25], which is made of nanomaterials and uses chemical dynamics in the materials as computations in RC. We particularly employ an in-material RC made from a sulfonated polyaniline network (SPAN). Since research on RC-based image recognition systems for hardware is still in its infancy, the contribution of this study must be significant.

II. Related works

A. Reservoir-based convolution

In the reservoir-based convolution, reservoirs are used as filters receiving a region of interest (ROI) from an image, as shown in Fig. 1. The ROI is fed into the reservoir line by line, and the reservoir updates its internal state as follows:

$$\boldsymbol{x}(t) = (1-\delta)\boldsymbol{x}(t-1) + \delta f(W_{\mathrm{i}}\boldsymbol{u}(t) + W_{\mathrm{r}}\boldsymbol{x}(t-1)), \quad (1)$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{N_{\mathrm{r}}}$ and $\boldsymbol{u}(t) \in \mathbb{R}^{N_{\mathrm{i}}}$ are the reservoir's internal state and an input vector (the one line from the ROI), respectively, and t indicates a discrete time step. $W_{\mathrm{i}} \in \mathbb{R}^{N_{\mathrm{r}} \times N_{\mathrm{i}}}$ and $W_{\mathrm{r}} \in \mathbb{R}^{N_{\mathrm{r}} \times N_{\mathrm{r}}}$ denote a weight connection matrix between the input and the reservoir and a recurrent connection matrix in the reservoir, respectively, (N_{i} and N_{r} represent the number of input and reservoir nodes). f indicates a nonlinear function and the hyperbolic tangent function is used for this. δ denotes the leak rate of the reservoir ($0 < \delta < 1$) that controls the updating speed of the reservoir. The second term in Eq. 1 has a relatively larger effect and causes the updating speed to increase if the leak rate is close to 1. Conversely, the updating speed slows down if the leak rate is close to 0.

The reservoir receiving ROIs can function as a feature extractor for specific spatial frequencies using that reservoir characteristic. The reservoir is insensitive to features with high spatial frequencies if the ROI is fed into a reservoir with a low leak rate because the reservoir update cannot keep up with the input change. Otherwise, a reservoir with a high leak rate is sensitive to features with high spatial frequencies.

The reservoir-based convolution consists of multiple reservoirs with multiple leak rates to extract features with various spatial frequencies, as shown in Fig. 2. An input image is divided into ROIs in the same manner as CNNs, each ROI is fed into all reservoirs, and the reservoirs update their states as Eq. (1). If the input $\boldsymbol{u}(t)$ is T steps ($T = 1, 2, \ldots, T$), all reservoir states that occur immediately after data feeding, $\boldsymbol{x}(T)$ are concatenated and used as outputs. Output feature maps are generated by independently feeding all ROIs into the reservoirs.

Figure 3 shows an example structure of a reservoir-based CNN consisting of two reservoir-based convolution layers, two max-pooling layers, and a linear layer. In this network, an input image is fed into the first reservoir-based convolution layer, and its output feature maps are down-sampled by the first max-pooling layer. The second reservoir-based convolution and the max-pooling layers process the outputs from the first max-pooling layer. Finally, the feature maps are flattened and fed into the linear layer and its outputs represent classification results.

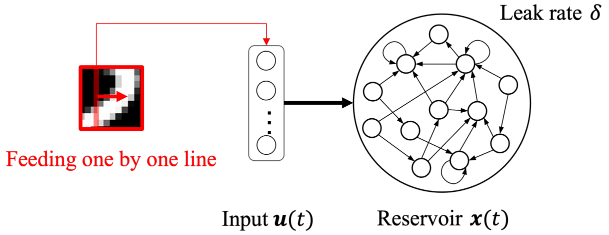


Fig. 1. Feeding image into the reservoir.

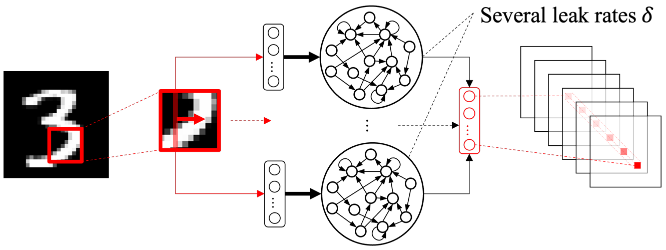


Fig. 2. Reservoir-based convolution.

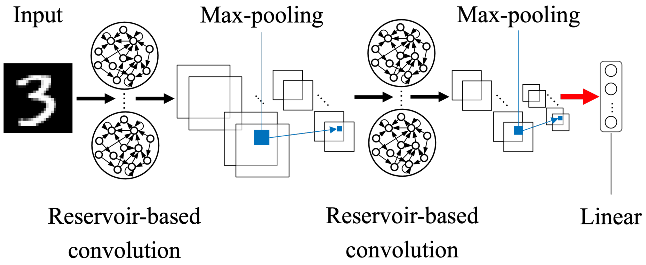


Fig. 3. Reservoir-based CNN.

The most significant feature of the reservoir-based CNN is that the weight connections in the reservoir-based convolution layers are fixed, and only the weight connection in the linear layer, indicated in the figure by a red arrow, has plasticity. Therefore, this network requires lower training costs than conventional CNNs.

B. *In-material reservoir computing*

A nanomaterial network system is a promising option among the various strategies for physical RC implementation because of the simple fabrication process. An electric carrier is a driving force that flows through the nanomaterial network. In the nanomaterial network, a complex carrier transport path is generated along with a network that includes nonlinear electrical behavior at junctions in nanosize and µs-ms timescale, which is interpreted as chemical dynamics. Thus, the chemical dynamics in the nanomaterial network have rich kinetics and spatio-temporal dynamics, which can produce high-dimensional nonlinear mapping in RC.

The SPAN network is a candidate for in-material RC. Figure 4 shows the structural formula of SPAN. SPAN is a derivative of polyaniline, which is one of the most famous conductive polymers [26]. SPAN forms polymer-aggregated string structures of approximately 5 nm width; therefore, a highly dense and complex network structure can be fabricated. SPAN has a protonated sulfonic functional group that performs a dopant function. In addition to other conductive polymers, polaron is the intrinsic carrier of SPAN. Furthermore, protons from the atmosphere are injected into the SPAN molecular chain. Polaron and ion in the SPAN network have different carrier mobilities (polaron: $0.25\ \mathrm{cm}^2\,\mathrm{V}^{-1}\,\mathrm{s}^{-1}$, ion: $3.6 \times 10^{-3}\ \mathrm{cm}^2\,\mathrm{V}^{-1}\,\mathrm{s}^{-1}$). Therefore, SPAN has rich and complex chemical dynamics owing to the combination of dual carriers.

A SPAN network RC device can be fabricated using the following process. A microscale metal 16 electrode pattern is fabricated on a SiO_2/Si substrate using the optical lithography method for creating input and output to the SPAN network. SPAN in an aqueous solution is drop-casted on metal electrodes to immobilize a network structure.

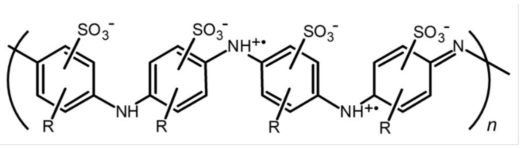


Fig. 4. Structural formula of sulfonated polyaniline.

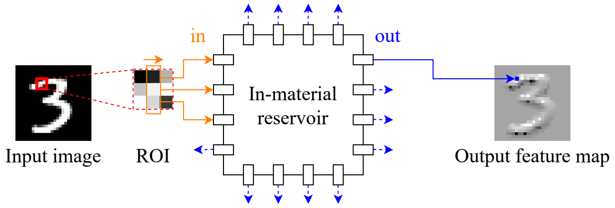


Fig. 5. Reservoir-based convolution circuit system using the in-material RC.

The waveform generation task, a typical benchmark task, is used to evaluate the RC performance of the SPAN network device. A sinusoidal wave input is applied to the RC device, and the 15 outputs are superimposed to target waves such as the cosine, triangle, sawtooth, and square. The accuracy of the predictions are more than 88%. These high accuracy levels imply that the SPAN network has a significant nonlinearity and high dimensionality as an RC device. Moreover, this in-material RC device achieves 60% accuracy of spoken-digit classification from the free-spoken-digit-dataset [27] with only 12 outputs, indicating that the SPAN network has a rich dynamic and nonlinear resource for RC.

III. Proposed method

This study proposes a reservoir-based convolution circuit system using an in-material RC, as shown in Fig. 5. The in-material RC device in the system has 16 input/output ports connected to the SPAN network and we use three of them for input ports and the remaining 13 ports for outputs. The device receives a 3×3-pixel ROI as a three-step time series of a three-dimensional vector, and the state of the output ports immediately after the feeding is used for the layer outputs. Each output port corresponds to a channel of the output feature map so that the number of output feature maps is 13 in the case of the device.

This study assumes that the input image value range is [0, 1], and we apply the image value to the device as a voltage, resulting in an input port range of [0V, 1V]. We discovered through an experiment that the range of output ports was approximately [-0.20 V, 0.20V]; therefore, we multiplied the output voltage by 5.0 to determine the value of the output feature maps.

We construct a reservoir-based CNN with one reservoir-based convolution layer and 13 feature map outputs, allowing us to implement the network using single in-material RC devices with 16 input/output ports, as shown in Fig. 6. The max-pooling and the linear layer computations are executed using the software.

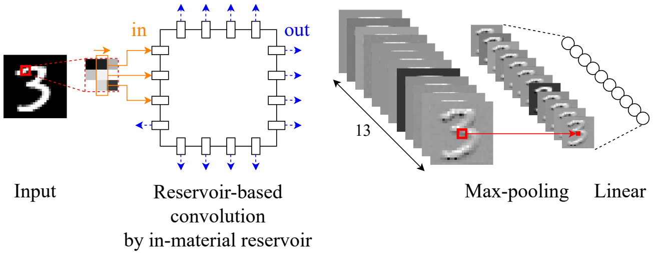


Fig. 6. Reservoir-based CNN using the in-material RC device.

IV. Experiments

A. Data feeding

Since the in-material RC device has a proper time constant where the signal is processed appropriately, an input signal frequency to the device affects the RC state changes. The reservoir state does not represent the entire feature of the time-series data if the input frequency is too low for the device because each input effect on the reservoir disappears at the following time step. Conversely, if the input frequency is too high for the device, the reservoir does not process the time-series data because the reservoir state change cannot keep up with the input change.

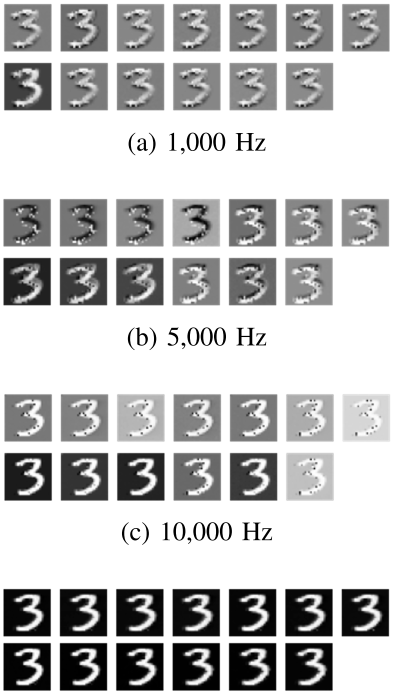
For the above reason, we investigated an appropriate input frequency for the in-material RC device in the reservoir-based convolution application. In this investigation, we changed the input frequency from 1,000 Hz to 50,000 Hz while verifying the RC device response, and we fed an image from the MNIST dataset to the RC device, as shown in Fig. 5.

Figure 7 shows the generated feature maps from the in-material RC devices, with the input frequency set to 1,000 Hz, 5,000 Hz, 10,000 Hz, and 50,000 Hz. Figures 7 (a) and (b) show the results when the input frequencies were set to 1,000 Hz and 5,000 Hz. In both cases, the reservoir-based convolution extracted the line's edges, and the feature maps varied. Conversely, in the case of 50,000 Hz, the generated feature maps were consistent and only slightly different from the input image. Although the generated feature maps in the case of 10,000 Hz varied, the edges of the lines could not be extracted accurately. Therefore, we concluded that 1,000 to 5,000 Hz was the appropriate input frequency for the in-material RC device.

B. Image classification task

To verify the performance of the reservoir-based CNN with the in-material RC, we conducted an image classification task using the MNIST dataset. Although the dataset consists of 60,000 training images and 10,000 test images of gray-scale handwritten digits, we used 1,000 training and 1,000 test images to simplify the device measurement.

We fed the training images into the reservoir-based CNN and computed an optimized synaptic weight connection of the linear layer by ridge regression using both the outputs of the max-pooling layer and the supervised signals that were one-hot vectors corresponding to the class labels. We then



(a) 1,000 Hz

(b) 5,000 Hz

(c) 10,000 Hz

(d) 50,000 Hz

Fig. 7. Generated feature maps from the in-material RC devices with respect to the input frequencies.

TABLE I
ACCURACY RATES OF THE MNIST DATASET CLASSIFICATION TASKS.

	Test accuracy rate
Linear model	0.744
Reservoir-based CNN by the ESN	0.877 ± 0.00957
Reservoir-based CNN by the in-material RC	0.817

fed the test images into the reservoir-based CNN and verified their accuracy rate. The parameter of the ridge regression λ was set to 1.0, and the input frequency to the in-material RC was set to 1,000 Hz. Table I compares the accuracy rate of the MNIST dataset classification task for a linear model, an ESN-implemented reservoir-based CNN, and a reservoir-based CNN with an in-material RC. Since the weight connections in the ESN were randomly initialized, the accuracy of the reservoir-based CNN by the ESN in the table represents the mean of ten trials.

V. Discussion

Table I demonstrates that the reservoir-based CNN by the in-material RC performed better than the linear model but less than the reservoir-based CNN by the ESN. We concluded that the cause was noise in the outputs produced by the circuit system, as shown in Fig. 7.

In the experiment of the image classification task, the input frequency was set to 1,000 Hz, causing the computation for a 3×3 ROI to take 3 ms. Therefore, it is estimated that using one in-material RC device, the computation time for all ROIs (676 ROIs in this case) will be at least 2.03 s. To accelerate this computation, we can implement the reservoir-based convolution in parallel processing using multiple in-material RC devices, resulting in a computation time of 3 ms maximum. Additionally, a higher input frequency can accelerate the computation, but accuracy verification is required.

The in-material RC is suitable for reservoir-based convolution from the viewpoint of the device characteristics. Although the in-material RC has a small memory capacity, the reservoir does not require a large memory capacity because the reservoir only receives a 3×3 ROI that is converted into a three-step time series of a three-dimensional vector.

VI. Conclusion

This study proposes a reservoir-based convolution circuit system using in-material RC devices made from the SPAN to develop an efficient image recognition system for edge AI. The experimental results showed that the proposed circuit system extracted features from images as the original convolutional operation, and the reservoir-based CNN on the in-material RC achieved an accuracy rate of 81.7% in the image classification task, while the ESN-based CNN and the linear model achieved 87.7% and 74.4%, respectively.

In this study, we used a single in-material RC device and a single input frequency to simplify the device measurement, which prevented the reservoir-based CNN from being able to extract features with various spatial frequencies as in the original method. In the future, we will investigate methods for extracting various features using in-material RC devices. Additionally, the max-pooling and linear layers were software-based rather than being implemented on the circuit. For a low-power implementation, we must implement these layers into an integrated circuit (IC) and measure its power consumption to show the effectiveness of the proposed method. Because of the limitation of the device measurement, we only conducted the experiment with 2,000 images. The IC implementation of the proposed network will overcome the limitation and we will be able to conduct an experiment using full of the MNIST dataset or other datasets to compare with the previous studies.

Acknowledgment

This paper is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and supported by JSPS KAKENHI Grant Number 21K14527, 21K21318, 22K17968, JSPS Core-to-Core Program Grant Number JPJSCCA20220006, JST CREST Grant Number JPMJCR21B5, and ACT-X Grant Number JPMJAX22K4. This work was technologically supported by Yamaguchi University and Kitakyushu Semiconductor Center under the “Advanced Research Infrastructure for Materials and nanotechnology in Japan (ARIM Japan)” of the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. Y. U. thanks to Asahi Kohsan Co., Ltd. for financial support through the Kitakyushu Foundation for the Advancement of Industry, Science, and Technology, Japan.

References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, p. 1527–1554, 2006.

[2] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, 2001.

[3] W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[5] P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, “Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics,” *Neural Networks*, vol. 126, pp. 191–217, 2020.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] N. Schaetti, M. Salomon, and R. Couturier, “Echo state networks-based reservoir computing for mnist handwritten digits recognition,” in *2016 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE Intternaional Conference on Embedded and Ubiquitous Computing (EUC) and 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, 2016, pp. 484–491.

[8] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017. [Online]. Available: https://arxiv.org/abs/1708.07747

[9] Z. Tong and G. Tanaka, “Reservoir computing with untrained convolutional neural networks for image recognition,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 1289–1294.

[10] Y. Yonemura and Y. Katori, “Image recognition model based on convolutional reservoir computing,” *The 34th Annual Conference of the Japanese Society for Artificial Intelligence*, 2020.

[11] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

[12] Q. An, K. Bai, L. Liu, F. Shen, and Y. Yi, “A unified information perceptron using deep reservoir computing,” *Computers and Electrical Engineering*, vol. 85, p. 106705, 2020.

[13] Y. Tanaka and H. Tamukoh, “Reservoir-based convolution,” *Nonlinear Theory and Its Applications, IEICE*, vol. 13, no. 2, pp. 397–402, 2022.

[14] K. Honda and H. Tamukoh, “A hardware-oriented echo state network and its fpga implementation,” *Journal of Robotics, Networking and Artificial Life*, vol. 7, pp. 58–62, 2020.

[15] M. L. Alomar, E. S. Skibinsky-Gitlin, C. F. Frasser, V. Canals, E. Isern, M. Roca, and J. L. Rosselló, “Efficient parallel implementation of reservoir computing systems,” *Neural Computing and Applications*, vol. 32, no. 7, pp. 2299–2313, 2020.

[16] N.-S. Huang, J.-M. Braun, J. C. Larsen, and P. Manoonpong, “A scalable echo state networks hardware generator for embedded systems using high-level synthesis,” in *2019 8th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2019, pp. 1–6.

[17] L. Loomis, N. McDonald, and C. Merkel, “An fpga implementation of a time delay reservoir using stochastic logic,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 4, pp. 1–15, 2018.

[18] M. L. Alomar, V. Canals, N. Perez-Mora, V. Martínez-Moll, and J. L. Rosselló, “Fpga-based stochastic echo state networks for time-series forecasting,” *Computational intelligence and neuroscience*, vol. 2016, 2016.

[19] H. Suzuki, J.-i. Imura, Y. Horio, and K. Aihara, “Chaotic boltzmann machines,” *Scientific reports*, vol. 3, no. 1, pp. 1–5, 2013.

[20] I. Kawashima, Y. Katori, T. Morie, and H. Tamukoh, “An area-efficient multiply-accumulation architecture and implementations for time-domain neural processing,” in *2021 International Conference on Field-Programmable Technology (ICFPT)*, 2021.

[21] K. Nakajima, “Physical reservoir computing—an introductory perspective,” *Japanese Journal of Applied Physics*, vol. 59, no. 6, p. 060501, may 2020.

[22] G. Van der Sande, D. Brunner, and M. C. Soriano, “Advances in photonic reservoir computing,” *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[23] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima *et al.*, “Neuromorphic computing with nanoscale spintronic oscillators,” *Nature*, vol. 547, no. 7664, pp. 428–431, 2017.

[24] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, “Information processing via physical soft body,” *Scientific reports*, vol. 5, no. 1, pp. 1–11, 2015.

[25] Y. Usami, B. van de Ven, D. G. Mathew, T. Chen, T. Kotooka, Y. Kawashima, Y. Tanaka, Y. Otsuka, H. Ohoyama, H. Tamukoh, H. Tanaka, W. G. van der Wiel, and T. Matsumoto, “In-materio reservoir computing in a sulfonated polyaniline network,” *Advanced Materials*, vol. 33, no. 48, p. 2102688, 2021.

[26] S. Shimizu, T. Saitoh, M. Uzawa, M. Yuasa, K. Yano, T. Maruyama, and K. Watanabe, “Synthesis and applications of sulfonated polyaniline,” *Synthetic Metals*, vol. 85, no. 1-3, pp. 1337–1338, 1997.

[27] Z. Jackson, “Free spoken digit dataset.” [Online]. Available: https://github.com/Jakobovski/free-spoken-digit-dataset