Spiking Neural Networks for Efficient Streaming Event Detection

G. William Chapman
Sandia National Laboratories
Albuquerque, NM, USA
Email: gwchapm@sandia.gov

Martha Gahl
Sandia National Laboratories
Albuquerque, NM, USA
Email: mtgahl@sandia.gov

William Wahby
Sandia National Laboratories
Albuquerque, NM, USA
Email: wswahby@sandia.gov

Frances S. Chance
Sandia National Laboratories
Albuquerque, NM, USA
Email: fschanc@sandia.gov

Sapan Agarwal
Sandia National Laboratories
Albuquerque, NM, USA
Email: sagarwa@sandia.gov

T. Patrick Xiao
Sandia National Laboratories
Albuquerque, NM, USA
Email: txiao@sandia.gov

***Abstract*—Real-time and efficient data processing is a requirement for edge-based applications such as space remote sensing, where more data is collected at sensors than can be buffered or stored. We present a neurally-inspired machine learning architecture that enables streaming detection of events of interest based on spatiotemporal patterns, while maintaining low complexity and binary activations. The architecture is compatible with low-power microelectronics for analog in-memory computing.**

***Keywords*—Remote Sensing; Neuromorphic Computing; Analog Computing**

I. Introduction

Satellite-based remote sensing systems use high-resolution focal plane arrays (FPAs) to monitor large areas of the Earth. The FPA generates far more data than can be transmitted to the ground, so a real-time response to perceived events or objects on Earth requires significant data reduction, achieved by data processing on or near the FPA. The more numerous the pixels, the greater the amount of processing that must fit within the satellite's size, weight and power (SWaP) constraints. Therefore, the ground resolution of these systems depends directly on the complexity of the algorithm and the energy efficiency of the processing hardware. We seek to develop computational approaches that can down-select FPA regions of interest in a streaming manner and require no or minimal buffering of past pixel values. We investigate compact and efficient neural networks for spatiotemporal event detection, and show how these can be implemented on low-power analog microelectronics for processing matrix-vector-multiplications (MVMs) and neuron functions. We evaluate a wide range of network architectures and show that convolutional recurrent networks[1] perform on par with larger networks such as gated recurrent units, or networks requiring explicit buffering of several frames. We further show that such networks can utilize a spiking activation function [2] to replace analog-to-digital conversion, further reducing power. Finally, we demonstrate via simulation that these networks are robust to errors arising from variability in SONOS flash memory devices[3].

II. Background

A. Neurally Inspired Processing for Remote Sensing

Animal visual systems have evolved to efficiently process large amounts of visual data and can rapidly extract important signals from visual information, allowing for quick decision-making. We propose solutions, fundamentally inspired by neurological signal processing and architecture, to solve event detection tasks where SWaP constraints preclude the use of more conventional hardware.

Standard event detection approaches tend to implement sequential spatial-then-temporal processing. This is only effective when there are spatial features within the event that change through time or spatial features unique to different signal types. In contrast, biological nervous systems possess local lateral interactions at every stage of processing, including nonlinear interactions as early as the retina [4]. These connections allow extraction of spatiotemporal patterns, as opposed to purely spatial feature extraction [5], and efficient filtering of redundant information. This enables systems to be lightweight while still being reliable. We implement this neurally inspired processing by incorporating convolutional recurrent neural networks (CRNNs) which contain local lateral interactions, as well as feedforward activity.

Biological neural systems are also highly energy efficient, owing in part to the temporally discrete, saltatory nature of inter-unit signaling. Research has demonstrated that replacing the real-valued activation functions of artificial neural networks with simple spiking neural network (SNN) activations can result in high performance using frameworks provided by the machine learning field [2]. When implementing directly as electronic circuits, SNNs achieve drastically reduced energy consumption compared to analog-to-digital conversion [6], partly due to the temporally sparse nature of communication. Beyond potential SWaP benefits, SNNs are inherently temporal in nature, and may allow efficient event detection without the explicit local interactions of CRNNs.

DISTRIBUTION STATEMENT A. Approved for public release: distribution is unlimited. Approval ID: <SAND2026-16543C>.

B. Analog Neural Networks

Universal computers such as graphics processing units (GPU) and field programmable gated arrays (FPGA) have high throughput and allow execution of arbitrary machine learning architectures. However, their SWaP consumption is often too high for edge-computing applications, especially when compute must be near physical sensors. An alternative is analog neural network accelerators [7], which utilize passive electronic elements to implement portions of machine learning algorithms in an energy-efficient manner [8]. Analog neural networks have shown minimal performance decreases from their digital counterparts [9] when trained to be aware of the inherent noise that exists in such devices. One platform of interest is crossbar arrays, which utilize memristors as programmable resistors to accelerate in-memory matrix-vector multiplication (MVM). While MVM is required by all ANNs, state-of-the-art event detection algorithms incorporate additional operations, such as Kalman filtering, particle filtering, or template matching [10], [11], [12].

C. Architecture Constraints of Edge Applications

Many common architectures used in object or event detection tasks are not feasible candidates for edge applications because of the intensive data movement requirements. Two-stage detection models, like Faster R-CNN, require multiple large networks or stored weights beyond what current arrays would allow [13]. Alternative single-stage detection models, such as YOLO [14] or single-shot detectors [15], do not detect with temporally changing data. Standard temporal tracking methods require simultaneous access to all frames of a video, precluding use in online detection tasks, and rely on static spatiotemporal filters rather than allowing dynamic integration of previous and current inputs. In the online detection tasks that we address here, a network may only access inputs from the current time step and any information explicitly maintained via internal states or recurrent connectivity. Transformer models, which are increasingly popular for sequence data, can incorporate spatial components as well [16]. However, such systems require nonlinear attention mechanisms, continual buffering of inputs via a shifting context trace, and other mechanisms all of which require additional auxiliary circuitry, introducing complexity for circuit design. Event detection on edge applications requires simple neural network architectures.

III. Methods

A. Task Description

We designed a temporal event detection task in which each trial consists of 100 sequentially presented 10-by-10 frames, and the network's task is to detect the presence of an event of interest on each frame. Because the events are spatial and temporal in nature, networks are trained with backpropagation-through-time techniques, and performance is measured on a frame-by-frame level. To simulate a semi-realistic scenario, events are superimposed on a parametrically generated background that has spatial frequency components that are typical of satellite imagery. Events include both targets and distractors. Targets are sigmoidal temporal signals which have an amplitude, onset time, rise over some period, and then plateau for the remainder of a trial. Distractors are similar, except they are Gaussian in time, decreasing in amplitude after reaching a peak. Both types of events are modeled as sub-pixel points, and convolved, along with the background, by a simulated optical point-spread function. Examples of the spatiotemporal profiles of events are shown in Figure 2. The maximum amplitude, rise time (for sigmoids) and standard deviation (Gaussians) are randomly drawn on each trial, according to Table I such that the amplitude or rise time alone are insufficient to distinguish targets from distractors. All experiments are performed on a 7,500 trial training dataset, and tested on a separately generated test dataset of the same size and characteristics.

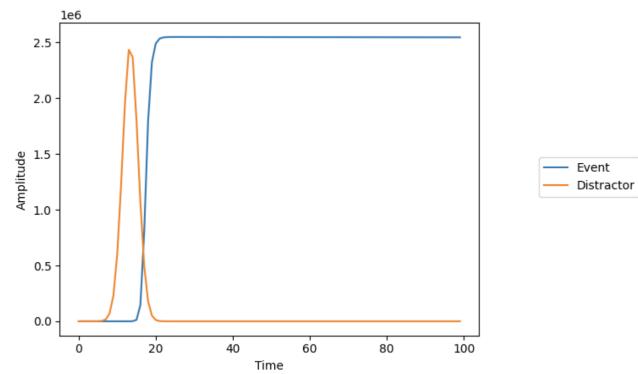

Fig. 1. Example event amplitudes throughout a trial. The blue line shows the shape of target events, which are sigmoidal. The orange line shows the shape of distractor events, which are Gaussian.

TABLE I
DISTRIBUTION OF PARAMETERS FOR TASK GENERATION, DRAWN RANDOMLY ON EACH TRIAL.

Parameter	Distribution	Default Minimum	Default Maximum	Signal Type
Location (x and y)	Bivariate Uniform	0	frame size	Gaussian and Sigmoidal
Amplitude	Uniform Exponent	$log_{10}(1000)$	5	Gaussian and Sigmoidal
Onset (frames)	Uniform	4	30	Gaussian and Sigmoidal
Rise	Uniform	0.5	10	Sigmoidal
Width	Uniform	0.5	10	Gaussian
Offset	Uniform	4	30	Gaussian

B. Network Architecture

We performed an architecture search for minimally complex models that can distinguish events and distractors. Models included convolutional (C), recurrent (R), and convolutional recurrent (CR) layers in a feedforward or locally-recurrent sequence, allowing streaming processing without buffering past observations. Convolutional Recurrent layers combine

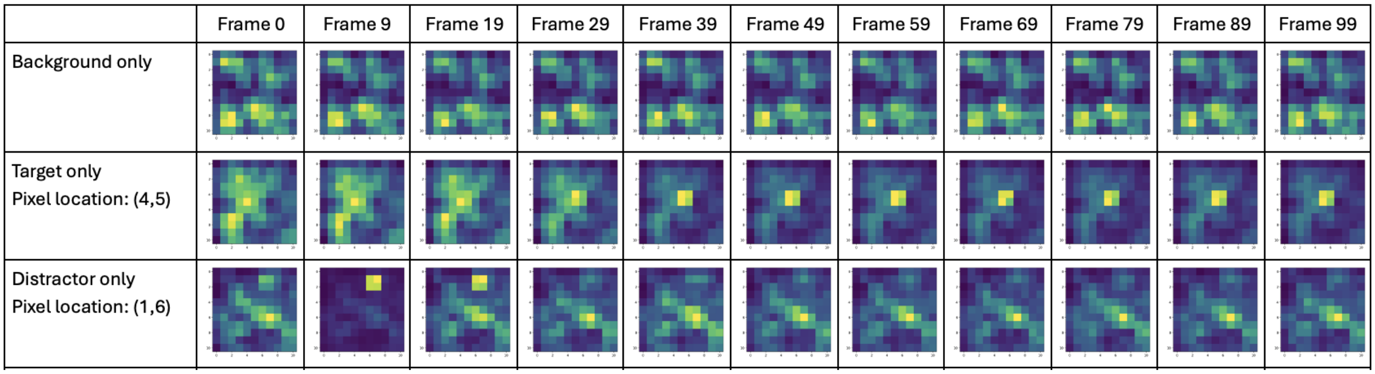

Fig. 2. Example trials for three conditions, normalized per-frame to enhance detail. **Top** Background images are randomly cluttered to obfuscate small SNR events. **Middle** Sigmoidal targets rise at a randomly chosen frame, and remain on for the remainder of a 100-frame trial. **Bottom** Gaussian distractors occur on a random frame and rise for a short time before reaching inflection (frame 10 here) and decaying.

spatial and temporal structure into a single layer, by the function:

$$Y(X, Y(t-1)) = F(W_{ih} * X + W_{hh} * (Y(t-1))) \quad (1)$$

Where Y is the output of the CR operation, F is the activation function, X is the feedforward input from the previous layer, and W_{ih} and W_{hh} are feedforward and recurrent convolutional kernels. The convolutional operator $(*)$, when applied to the feedforward input (X) allows spatial features from the input to be extracted. When convolution is applied to the recurrent input (Y) it allows the extraction of spatiotemporal features [1].

C. Spiking Neural Networks

In addition to standard activation functions (rectified linear for convolutional, hyperbolic tangent for recurrent), we evaluated the performance of spiking activations. Spiking activations utilize a local 'sub-threshold' state on a pixel-by-channel basis, which functions as a low-pass filter of past activity, combined with a binary threshold function, as demonstrated by equation 2. Spiking activation functions can have lower energy than standard activations which require higher bit-resolution, while still allowing trainability for recurrent topologies that is impossible with standard binary activation functions [17]. Performance for both floating-point and spiking activation architectures is summarized in Table II.

$$\begin{aligned} \tau_v \frac{dv_L(t)}{dt} &= -v_L(t) + \sum_{n \epsilon A} W_{nL} S_n(t) \\ S_L(t) &= v_L(t) \geq 1 \\ v_L(t+1) &= \begin{cases} 0, & \text{if } v_L(t) + \frac{dv_L(t)}{dt} > 1 \\ v_L(t) + \frac{dv_L(t)}{dt}, & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

D. Hyperparameter & Training Procedures

Unless otherwise noted, all initializations and hyperparameters follow PyTorch defaults [18]. With the layer types above, our hyperparameter search focused on several properties and values: loss function, weight decay, learning rate, and confidence threshold for positive identification.

Networks were optimized to minimize cross-entropy loss, calculated per frame and averaged over each trial. All networks were optimized with backpropagation-through-time (BPTT) to account for the temporal nature of the task and a subset of the networks. For spiking networks, surrogate gradient descent was performed with the SuperSpike [19] function applied to the membrane potential. Optimization was performed with stochastic gradient descent using a weight decay of 0.1 and an initial learning rate of 0.0001. Gradient clipping was utilized to reduce oscillations in the BPTT-based optimizations.

The results section contains several performance metrics. Accuracy and false-positive rates are based on a threshold of 0.5, and are reported for both frame-level and trial-level statistics. Frame-level accuracy is the overall agreement between ground-truth and network classification on a frame-by-frame level averaged across the test datasets. Trial accuracy measures whether networks classified any frame after ground-truth onset as containing an event. If the network classifies a frame before ground-truth onset as containing an event this is counted as a false positive. We also present the average delay, in frames, between ground-truth event onset and network detection. A lower delay indicates that the network was able to detect events more quickly, and therefore at a lower signal-to-noise ratio compared to a higher delay. The use of trial-averaged cross-entropy implicitly optimizes for all of these measurements, but the ideal balance of delay versus false-positives will depend on other task constraints or relative importance of fast detection.

IV. Results

A. Architecture Performance

Table II summarizes performance metrics for a sequential convolutional-then-recurrent (C-R) networks and networks with integrated convolutional-recurrent (CR) layers, with floating point, binary, and spiking activation functions. For floating point, overall accuracy was similar for C-R and CR networks, but with a much lower delay to detection for CR networks. For binary activation networks, accuracy of CR networks was near

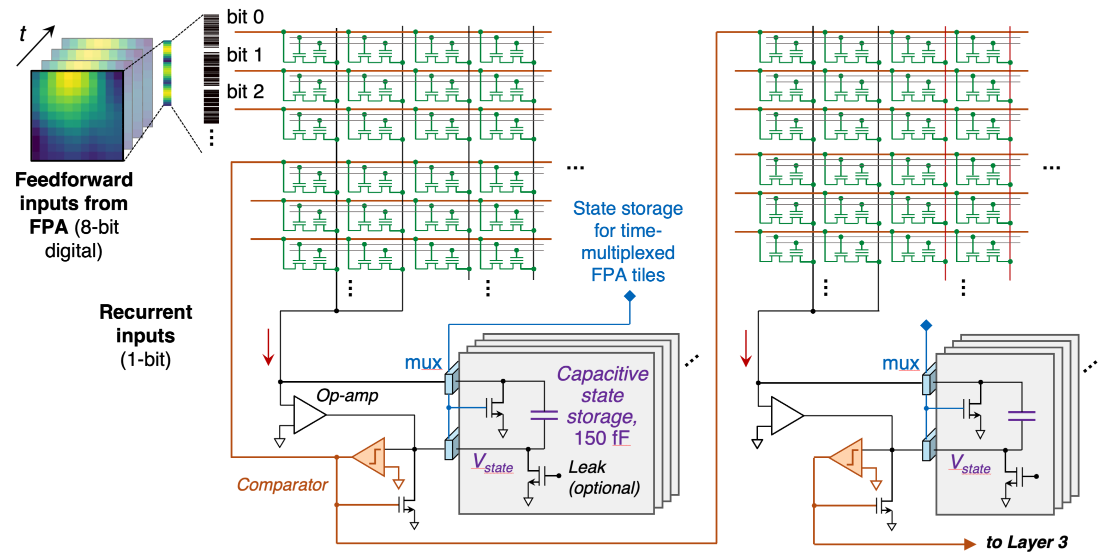

Fig. 3. Possible analog hardware implementation of spiking network. Real-valued weights are embedded in crossbar memory arrays, while state and activations are placed at the bottom of each column. The spiking neuron block replaces a multi-bit analog-to-digital converter, enabling significant energy savings. After the input layer bit-slicing is not longer required, as activations are binary.

chance, consistent with prior observations that binary activations are incompatible with recurrent architectures [17], and delay was significantly higher than floating point CR networks. Spiking neural networks performed closer to the floating point networks than to binary activation networks, with sequential C-R networks having an increased false positive rate. SNNs with CR architectures had accuracy near that of floating point, with an increased delay due to the intrinsic time constant of the underlying dynamics in equation 2. Overall, these results indicate that the Spiking CR architecture is the highest performing network with the potential for energy efficient implementations.

TABLE II
PERFORMANCE ACROSS ARCHITECTURES AND ACTIVATION FUNCTIONS

Architecture	Accuracy	False Positive Rate		Delay
		W/Distractors	W/O Distractors	(Frames)
C-R (Floating)	0.811	0.206	0.010	15.12
CR (Floating)	0.801	0.237	0.040	4.07
C-R (Binary)	0.729	0.678	0.407	11.36
CR (Binary)	0.670	0.737	0.303	16.89
C-R (Spiking)	0.721	0.375	0.053	14.22
CR (Spiking)	0.803	0.230	0.018	12.14

B. Hardware Simulation

To evaluate the potential for deployment to analog hardware, we simulated network performance in response to realistic noisy simulations. We began with the final trained spiking CR network outlined above, and used CrossSim [20] to simulate realistic SONOS write errors on overall network performance. The hardware-converted network was then fine-tuned by hardware-aware training (HAT). In the HAT process, the noisy forward pass was performed using simulated analog MVM operations in CrossSim, and backward optimization was performed on idealized linear MVMs. The pretrained digital networks were trained under HAT for 50 epochs, using the same training procedure as described in the methods.

TABLE III
NETWORK PERFORMANCE ON POTENTIAL HARDWARE PLATFORMS

Frame Accuracy	Accuracy	False Positive Rate		Delay
		W/Distractors	W/O Distractors	(Frames)
Digital	0.803	0.230	0.018	12.14
Analog (no HAT)	0.775	0.459	0.032	11.86
Analog (HAT)	0.811	0.258	0.018	12.15

Performance of these experiments is summarized in Table III, with 'digital' representing the baseline performance of the spiking CR architecture from Table II. Direct translation from digital to analog, without hardware-aware training, resulted in a moderate decrease in accuracy, primarily through an increased false positive rate when tested with distractors. After hardware-aware training, accuracy is close to digital baseline performance, with a modest increase in false positive rate and delay. Overall, these results show that SNN architectures can be trained to be resilient to realistic levels of analog hardware noise.

V. DISCUSSION

In this work we investigate neurally inspired algorithms for robust spatiotemporal event detection. By introducing distractor signals, we can ensure that the network is learning about the signal properties themselves and not using artifacts of the dataset to achieve high performance. This allows the network to be more readily transferable to data outside the training distribution.

We found that simultaneous temporal and spatial processing of signals resulted in the highest performance, lowest false-positive rate, and lowest latency in event detection. These results are all desirable for FPA applications where there is limited throughput for readout or buffer of frames, and therefore require a local spatiotemporal processing to detect candidate tiles which may contain information of interest. In order to put these networks as close to the sensor as possible, we investigated binary activation and spiking implementations of the spatiotemporal networks and found that, consistent with previous approaches, binary activation recurrent networks fail to train. The convolutional recurrent spiking neural network is therefore the most plausible candidate for near-sensor streaming event detection. Finally, through hardware-aware training, we verified that such networks can be robust to the intrinsic device noise of analog crossbars.

Overall these results demonstrate that integrated spiking neural networks with analog matrix-vector multiplication circuit, as illustrated in Figure 3, are a promising candidate for streaming event detection under hardware constraints.

ACKNOWLEDGMENT

[1] G. William Chapman and Martha Gahl contributed equally to this work.

The authors acknowledge financial support from the Laboratory Directed Research and Development Program (Projects *229363*, *240547*) at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan.

REFERENCES

[1] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving Deeper into Convolutional Networks for Learning Video Representations," *arXiv preprint arXiv:1511.06432*, 2015.

[2] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training Spiking Neural Networks Using Lessons from Deep Learning," *Proceedings of the IEEE*, 2023.

[3] V. Agrawal, T. P. Xiao, C. H. Bennett, B. Feinberg, S. Shetty, K. Ramkumar, H. Medu, K. Thekkekara, R. Chettuvetty, S. Leshner, Z. Luzada, L. Hinh, T. Phan, M. Marinella, and S. Agarwal, "Subthreshold operation of SONOS analog memory to enable accurate low-power neural network inference," in *2022 International Electron Devices Meeting (IEDM)*. IEEE, 2022, pp. 21–7.

[4] F. S. Chance and S. G. Cardwell, "Shunting Inhibition as a Neural-Inspired Mechanism for Multiplication in Neuromorphic Architectures," in *Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference*, 2023, pp. 41–46.

[5] W. Lotter, G. Kreiman, and D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," Feb. 2017.

[6] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses with Spike-Timing Dependent Plasticity," *IEEE transactions on neural networks*, vol. 17, no. 1, pp. 211–221, 2006.

[7] Q. Xia and J. J. Yang, "Memristive Crossbar Arrays for Brain-Inspired Computing," *Nature Materials*, vol. 18, no. 4, pp. 309–323, Apr. 2019.

[8] S. Agarwal, T.-T. Quach, O. Parekh, A. H. Hsia, E. P. DeBenedictis, C. D. James, M. J. Marinella, and J. B. Aimone, "Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and Its Application to Sparse Coding," *Frontiers in neuroscience*, vol. 9, p. 484, 2016.

[9] T. P. Xiao, B. Feinberg, C. H. Bennett, V. Prabhakar, P. Saxena, V. Agrawal, S. Agarwal, and M. J. Marinella, "On the Accuracy of Analog Neural Network Inference Accelerators," *IEEE Circuits and Systems Magazine*, vol. 22, no. 4, pp. 26–48, 2022.

[10] S. Afshar, A. P. Nicholson, A. Van Schaik, and G. Cohen, "Event-Based Object Detection and Tracking for Space Situational Awareness," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 15 117–15 132, Dec. 2020.

[11] T. J. Ma and R. J. Anderson, "Remote Sensing Low Signal-to-Noise-Ratio Target Detection Enhancement," *Sensors*, vol. 23, no. 6, p. 3314, 2023.

[12] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-Temporal Video Autoencoder with Differentiable Memory," Sep. 2016.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single Shot Multibox Detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[16] A. H. d. O. Fonseca, E. Zappala, J. O. Caro, and D. van Dijk, "Continuous Spatiotemporal Transformers," Jul. 2023.

[17] G. W. Chapman, C. Teeter, S. Agarwal, T. P. Xiao, P. Hays, and S. S. Musuvathy, "Biological Dynamics Enabling Training of Binary Recurrent Networks," in *2024 Neuro Inspired Computational Elements Conference (NICE)*, Apr. 2024, pp. 1–7.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019.

[19] F. Zenke and S. Ganguli, "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks," *Neural Computation*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.

[20] T. P. Xiao, C. H. Bennett, B. Feinberg, M. J. Marinella, and S. Agarwal, "CrossSim: Accuracy Simulation of Analog in-Memory Computing."