frontiers | Frontiers in Neuroinformatics

TYPE Original Research
PUBLISHED 04 May 2023
DOI 10.3389/fninf.2023.1144301

OPEN ACCESS

EDITED BY
Bert Offrein,
IBM Research - Zurich, Switzerland

REVIEWED BY
Michael Wynn Hopkins,
The University of Manchester, United Kingdom
Leslie Samuel Smith,
University of Stirling, United Kingdom

*CORRESPONDENCE
Md Abdullah-Al Kaiser
✉ mdabdull@usc.edu

†These authors have contributed equally to this work

RECEIVED 14 January 2023
ACCEPTED 13 April 2023
PUBLISHED 04 May 2023

CITATION
Kaiser MA-A, Datta G, Wang Z, Jacob AP, Beerel PA and Jaiswal AR (2023) Neuromorphic-P^{2}M: processing-in-pixel-in-memory paradigm for neuromorphic image sensors.
Front. Neuroinform. 17:1144301.
doi: 10.3389/fninf.2023.1144301

Neuromorphic-P^{2}M: processing-in-pixel-in-memory paradigm for neuromorphic image sensors

Md Abdullah-Al Kaiser[1,2*†], Gourav Datta[1†], Zixu Wang[1], Ajey P. Jacob[2], Peter A. Beerel[1,2] and Akhilesh R. Jaiswal[1,2]

[1]Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, United States, [2]Information Sciences Institute, University of Southern California, Los Angeles, CA, United States

Edge devices equipped with computer vision must deal with vast amounts of sensory data with limited computing resources. Hence, researchers have been exploring different energy-efficient solutions such as near-sensor, in-sensor, and in-pixel processing, bringing the computation closer to the sensor. In particular, in-pixel processing embeds the computation capabilities inside the pixel array and achieves high energy efficiency by generating low-level features instead of the raw data stream from CMOS image sensors. Many different in-pixel processing techniques and approaches have been demonstrated on conventional frame-based CMOS imagers; however, the processing-in-pixel approach for neuromorphic vision sensors has not been explored so far. In this work, for the first time, we propose an asynchronous non-von-Neumann analog processing-in-pixel paradigm to perform convolution operations by integrating *in-situ* multi-bit multi-channel convolution inside the pixel array performing analog multiply and accumulate (MAC) operations that consume significantly less energy than their digital MAC alternative. To make this approach viable, we incorporate the circuit's non-ideality, leakage, and process variations into a novel hardware-algorithm co-design framework that leverages extensive HSpice simulations of our proposed circuit using the GF22nm FD-SOI technology node. We verified our framework on state-of-the-art neuromorphic vision sensor datasets and show that our solution consumes $\sim$ 2$\times$ lower backend-processor energy while maintaining almost similar front-end (sensor) energy on the IBM DVS128-Gesture dataset than the state-of-the-art while maintaining a high test accuracy of 88.36%.

KEYWORDS

neuromorphic, processing-in-pixel-in-memory, convolution, address event representation, hardware-algorithm co-design, DVS gesture

1. Introduction

Today's widespread video acquisition and interpretation applications [e.g., autonomous driving (Beltrán et al., 2020), surveillance (Xie et al., 2021), object detection (Jiao et al., 2022), object tracking (Wu et al., 2021), and anomaly detection (Mansour et al., 2021)] are fueled by CMOS image sensors (CIS) and deep learning algorithms. However, these computer vision systems suffer from energy inefficiency and throughput bottlenecks (Chai, 2020) that stem from the transmission of a high volume of data between the sensors at the edge and processors in the cloud. For example, smart glasses (e.g., Meta AR/VR glasses,

Google classes, etc.) drain the battery within 2–3 h when used for intensive computer vision tasks (LiKamWa et al., 2014). Although significant technological and system-level advancements exist in both CMOS imagers (Maheepala et al., 2020) and deep neural networks (Goel et al., 2020), the underlying energy inefficiency arises due to the physical separation of sensory and processing hardware. Hence, developing novel energy-efficient hardware for resource-constrained computer vision applications has attracted significant attention in the research community.

Many researchers implement the first few computation tasks of machine vision applications close to the sensor to reduce the energy consumption of massive data transfer (Zhou and Chai, 2020). These approaches can be categorized into three types (1) near-sensor processing, (2) in-sensor processing, and (3) in-pixel processing. The near-sensor processing approach places a digital signal processor or machine learning accelerator close to the sensor chip. In Pinkham et al. (2021), a dedicated near-sensor processor led to a 64.6% drop in inference energy for MobileNetV3. In Eki et al. (2021), a 3D stacked system consisting of a CNN inference processor and a back-side illuminated CMOS image sensor demonstrated an energy efficiency of 4.97 TOPS/W. Near-sensor computing can improve energy efficiency by reducing the data transfer cost between the sensor chip and the cloud/edge processor; however, the data traffic between the sensor and near-sensor processor still consumes significant amounts of energy.

In contrast, the in-sensor approach utilizes an analog or digital signal processor at the periphery of the sensor chip. For instance, RedEye (LiKamWa et al., 2016) uses analog convolution processing before the sensor's analog-to-digital conversion (ADC) blocks to obtain a 5.5× reduction in sensor energy. Moreover, a mixed-mode in-sensor tiny convolution neural network (CNN) (Hsu et al., 2022) yielded a significant reduction in bandwidth and, in particular, reduced power consumption associated with the ADC. To fully remove the ADC energy overhead, Chen et al. (2019) processed the raw analog data from the CMOS image sensor using an on-chip completely analog binary neural network (BNN) that leverages switched capacitors. Using energy-efficient analog computing was also explored in Ma et al. (2019), which proposes a novel current-mode analog low-precision BNN. Furthermore, SleepSpotter (Lefebvre et al., 2021) implemented energy-efficient current-domain on-chip MAC operations. Nevertheless, this solution still requires the potentially-compressed raw analog data to be streamed through column-parallel bitlines from the sensor nodes to the peripheral processing networks. In general, these in-sensor approaches significantly reduce the energy overhead of analog-to-digital converters; however, they still suffer from the data transfer bottleneck between the sensor and peripheral logic.

On the other hand, the in-pixel processing approach integrates computation capabilities inside the pixel array to enable early processing and minimize the subsequent data transmission. For instance, a low-voltage in-pixel convolution operation has been proposed in Hsu et al. (2020) that utilizes a current-based digital-to-analog converter (DAC) to implement weights and pulse-width-modulated (PWM) pixels. Moreover, a single instruction multiple data (SIMD) pixel processor array (PPA) (Bose et al., 2020) can perform parallel convolution operations within the pixel array by storing the weights of the convolution filters in registers within the in-pixel processing elements. In addition, the direct utilization of the photodetector current to compute the binary convolution can yield 11.49 TOPS/W energy efficiency (Xu et al., 2020). Furthermore, Xu et al. (2021) performs classification tasks on the MNIST dataset by generating the in-pixel MAC results of the first BNN layer and exhibits 17.3 TOPS/W energy efficiency. In addition, a processing-in-pixel-in-memory paradigm for CIS reported an 11× energy-delay product (EDP) improvement on the Visual Wake Words (VWW) dataset (Datta et al., 2022c). Follow-up works by the same authors have demonstrated 5.26× and 3.14× reduction in energy consumption on hyperspectral image recognition (Datta et al., 2022e) and multi-object tracking in the wild (Datta et al., 2022d), respectively. In summary, due to the embedded pixel-level processing elements, the in-pixel processing approach can outperform energy and throughput compared to in-sensor and near-sensor processing solutions.

Most of the research works on different energy-efficient CIS approaches (near-sensor, in-sensor, and in-pixel processing) are focused on conventional frame-based imagers. However, many researchers are now exploring the use of event-driven neuromorphic cameras or dynamic vision sensors (DVS) (Lichtsteiner et al., 2008; Leñero-Bardallo et al., 2011) for different neural network applications, including autonomous driving (Chen et al., 2020), steering angle prediction (Maqueda et al., 2018), optical flow estimation (Zhu et al., 2018), pose re-localization (Nguyen et al., 2019), and lane marker extraction (Cheng et al., 2020), due to their energy, latency, and throughput advantages over traditional CMOS imagers. The DVS pixel generates event spikes based on the change in light intensity instead of sensing the absolute pixel-level illumination in conventional CMOS imagers. Thus, DVS pixels filter out the redundant information from a visual scene and produce sparse asynchronous events. These sparse events are communicated off-chip using the address event link protocol (Lin and Boahen, 2009). By avoiding the analog-to-digital conversion of the absolute pixel intensity and frame-based sensing method, DVS exhibits higher energy efficiency, lower latency, and higher throughput than frame-based alternatives. Moreover, the dynamic range of the DVS pixel is higher than the conventional CMOS imagers; hence, the DVS camera can adapt to the illumination level of the scene due to its logarithmic receptor. These advantages motivate a paradigm shift toward neuromorphic vision sensors for vision-based applications.

These DVS cameras are often coupled with spiking convolution neural networks (CNN) that natively accept asynchronous input events. Traditionally, time is decomposed into windows, and the number of spikes that occur in each time window is accumulated independently for each pixel creating multi-bit inputs to a spiking CNN. The first spiking CNN layer thus consists of digital MAC operations (not accumulations because the input is multi-bit instead of binary), unlike the subsequent spiking CNN layers that consist of more energy-efficient accumulations that operate on spikes (Datta and Beerel, 2022; Datta et al., 2022b). To improve the energy efficiency of such a DVS system, this paper explores in-pixel processing by performing MAC operations in the analog domain within the pixel array. In particular, we have developed a novel energy-efficient neuromorphic processing-in-pixel-in-memory (P^2M) computing paradigm in which we implement the first spiking CNN layer using embedded transistors that model the multi-bit multi-channel weights and enable massively parallel

in-pixel spatio-temporal MAC operations. Because the DVS event spikes are asynchronous in nature, we perform the multiply operation by accumulating the associated weight each time a pixel event occurs. We threshold the accumulated value at the end of each time window to produce a binary output activation and reset the accumulator in preparation for the next time window. To support multiple input filters operating on individual pixels, we parallelize this operation and simultaneously operate on all channels (and all pixels). This charge-based in-pixel analog MAC operation exhibits higher energy efficiency than its digital off-chip counterpart. Moreover, the sparse binary output activations are communicated utilizing a modified address-event representation (AER) protocol, preserving the energy benefit of the workload sparsity. In addition, we have developed a hardware-algorithm co-design framework incorporating the circuit's non-linearity, process variation, leakage, and area consideration based on the GF22nm FD-SOI technology node. Finally, we have demonstrated the feasibility of our hardware-algorithm framework utilizing state-of-the-art neuromorphic event-driven datasets (e.g., IBM DVS128-Gesture, NMNIST) and evaluated our approach's performance and energy improvement. We incur a ∼5% accuracy drop in these datasets because our charge-based P^{2}M approach does not capture the conventional notion of membrane potential for the first CNN layer. This lack of membrane potential is due to the limited time a passive analog capacitor can effectively store charge without significant leakage. However, this problem can be mitigated using non-volatile memories (Jaiswal and Jacob, 2021) that we plan to explore in our future work.

The key contributions of our work are as follows:

1. We propose a novel neuromorphic-processing-in-pixel-in-memory (Neuromorphic-P^{2}M) paradigm for neuromorphic image sensors, wherein multi-bit pixel-embedded weights enable massively parallel spatio-temporal convolution on input events inside the pixel array.
2. Moreover, we propose non-von-Neumann charge-based energy-efficient in-pixel asynchronous analog multiplication and accumulation (MAC) units and incorporate the non-idealities and process variations of the analog convolution blocks into our algorithmic framework.
3. Finally, we develop a hardware-algorithm co-design framework considering hardware constraints (non-linearity, process variations, leakage, area consideration), benchmark the accuracy, and yield a $\sim 2\times$ improvement in backend-processor energy consumption on the IBM DVS128-Gesture dataset with a $\sim$ 5% drop in test accuracy.

The remainder of the paper is organized as follows. Section 2 describes the circuit implementation, operation, and manufacturability of our proposed Neuromorphic-P^{2}M approach. Section 3 explains our hardware-algorithm co-design approach and hardware constraints on the first layer of the neural network model. Section 4 demonstrates our experimental results on two event-driven DVS datasets and evaluates the accuracy and performance metrics. Finally, Section 5 presents the concluding remarks.

2. P^{2}M circuit implementation

This section presents the critical hardware innovations and implementation of our proposed neuromorphic-P^{2}M approach. Figure 1 illustrates the representative chip stack and computing flow for the first convolution layer utilizing our proposed neuromorphic-P^{2}M architecture. The top die consists of DVS pixels and generates ON (OFF) events based on the increase (decrease) in input light contrast level. A DVS pixel consists of a logarithmic receptor, source-follower buffer, capacitive-feedback difference amplifier, and two comparators (Lichtsteiner et al., 2008; Leñero-Bardallo et al., 2011; Son et al., 2017). The generated events (ON and OFF) per pixel are communicated to the bottom die via pixel-level hybrid Cu-to-Cu interconnects. The bottom die contains the weights and energy-efficient charge-based analog convolution blocks. Each DVS pixel's output channel (ON-channel and OFF-channel) is connected to a transistor in the bottom die that implements a multi-bit weight (e.g., $w_{1,ON}$, $w_{1,OFF}$, etc.) to perform the multiplication (e.g., $I_{1,ON} \times w_{1,ON}$, $I_{1,OFF} \times w_{1,OFF}$, etc.) operation. The positive and negative weights are implemented by utilizing the pMOS and nMOS transistors, respectively. Each kernel (corresponding to the filter of the spiking CNN model) accumulates its weighted multiplication of input events on an analog memory (capacitor) asynchronously when an ON or OFF event occurs in the input DVS pixel. As the input spikes are binary, the accumulation voltage either steps up (positive weight) or down (negative weight) by an amount, depending on the weight values. The accumulation continues for a fixed time period (simulation time length for each event stream of our neural network model), and after that, the summed voltage is compared with the threshold (using a comparator or skewed inverter) to generate the output activation signal (e.g., O_{ACT}) of each kernel for the next layer. A similar computing flow is used across the different kernels throughout the sensor array.

The operations of our proposed neuromorphic-P^{2}M can be divided into three phases. These are:-

1. Reset Phase: During the reset phase, the accumulation capacitor of each kernel is precharged to $0.5V_{DD}$ so that the accumulation voltage can step up or down within the supply rail depending on positive or negative weights, respectively.
2. Convolution Phase: In the convolution phase, the multi-bit weight-embedded pixels and the accumulation capacitor of each kernel perform multiplication and accumulation (MAC) operations in the analog domain for a fixed period of time. After that, the final accumulated voltage of each kernel is compared with a threshold voltage to (potentially) generate the output activation spike for the next layer.
3. Read Phase: Finally, during the read phase, the output activations of different kernels are sequentially read utilizing the asynchronous Address-Event Representation (AER) read scheme.

More details on each step, including their hardware implementations, will be explained below.

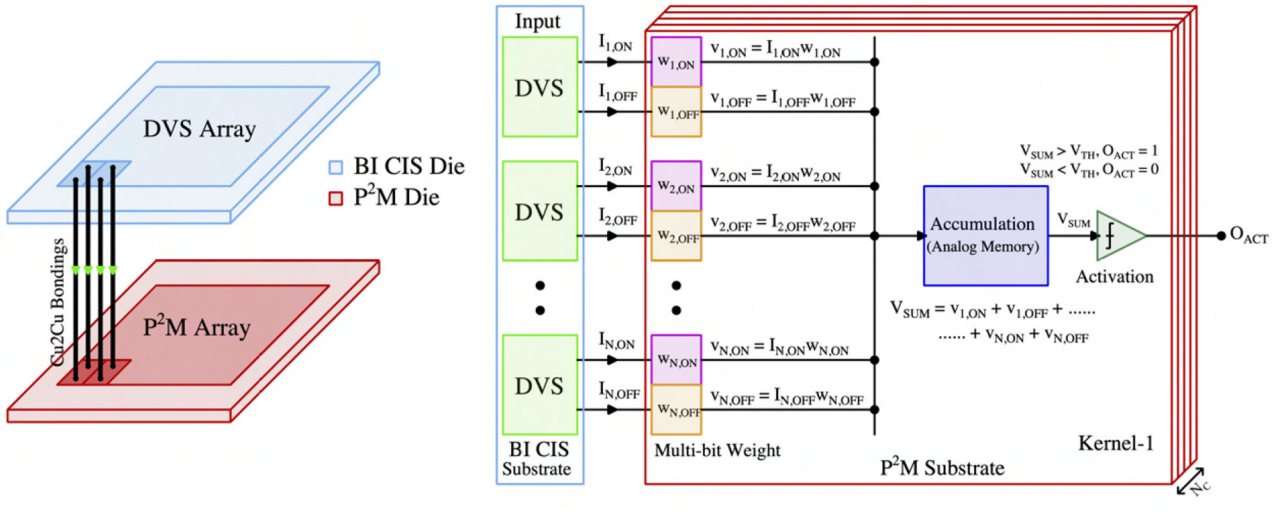

FIGURE 1
The representative 3D chip stack and computing flow diagram of our proposed neuromorphic-P^{2}M architecture.

2.1. Multi-bit weight embedded pixels

As illustrated in Figure 2, positive and negative weights of the first spiking CNN layer have been implemented by utilizing pMOS and nMOS transistors connected with supply voltages V_{DD} and ground, respectively. For a positive (negative) weight, the voltage across the kernel's capacitor (C_K) charges (discharges) from $0.5V_{DD}$ to V_{DD} (ground) as a function of weight values and the number of input DVS events. The weight values can be tuned by varying the driving strength ($\frac{W}{L}$ ratio) of the weight transistors (M_W). A high-V_T pMOS in positive weight implementation (nMOS in negative weight implementation) (M_{EN}) is activated during the convolution phase to enable the multiplication and accumulation operations on the kernel's capacitor (C_K) and remains off during the reset phase. The weight transistor (M_W) is chosen to have a high-V_T to limit the charging (for positive weight) or discharging (for negative weight) current to avoid capacitor saturation. Moreover, each DVS pixel includes a delayed self-reset circuit (consisting of a current-starved inverter chain and AND gate) to prevent voltage saturation on the capacitor (C_K) by limiting the event pulse duration. A switch transistor (M_{SW}) controlled by the DVS event spike is used to isolate the kernel's capacitor (C_K) from the weight transistor (M_W) to reduce the leakage. The switching transistor (M_{SW}) will be activated only when there are input DVS spikes, hence, ensuring the asynchronous MAC operation on the kernel's capacitor (C_K). Furthermore, to reduce the leakage, a kernel-dependent (as leakage is a function of transistor's geometry, hence, leakage amount is dependent on the kernel's weights) the current source (I_{NULL}) is connected with the accumulation capacitor (C_K) that flows in the opposite direction of the leakage current to nullify the leaky behavior of the capacitor. The number of weight transistors associated with a kernel depends on the size of the kernel (e.g., for a kernel size of 3 × 3, there will be a total of 18 weight transistors considering the ON and OFF-channel). Each kernel's weight transistors are connected to one accumulation capacitor (C_K).

Note that the weights cannot be re-programmed after manufacturing. However, it is common to use pre-trained weights for the first few layers as low-level feature extractors in modern neural network models (Jogin et al., 2018). Hence, the fixed weights of our proposed architecture do not limit its application for a wide range of machine-vision tasks. Moreover, we can also replace the transistor by utilizing a non-volatile memory device [e.g., Resistive Random Access Memory (RRAM), Phase Change Memory (PCM), Magnetic Random Access Memory (MRAM)] to add reconfigurability in our neuromorphic-P^{2}M approach.

To incorporate the circuit's non-ideality in our algorithmic model, we have simulated the output characteristics of the positive and negative weights for the different numbers of input event spikes using the GF 22nm FD-SOI node. Figure 3 represents the output voltage change on the accumulation capacitor (ΔV_{OUT}) as a function of the normalized weight transistor's $\frac{W}{L}$ ratios and different numbers of input event spikes. The figures show that the accumulated voltage can step up (for positive weights) and down (for negative weights), and the size of the step is dependent on the weight transistor's $\frac{W}{L}$ ratio. However, the step size dependency is non-linear, and the non-linearity is larger when the weights are large, and the pre-step voltage is close to the supply rails. This can be attributed to the fact that the weight transistors (M_W) enter the triode region when their drain-to-source voltage is low, causing the charging (discharging) current to drop compared to the typical saturation current. However, the number of input events is sparse for the DVS dataset, and having large weight values for all the weights in a kernel is highly unlikely for a neural network model. Hence, the weight transistors' non-linear characteristics do not cause significant accuracy issues in our algorithmic model. Besides, the circuit's asymmetry due to utilizing different types of transistors (pMOS for positive weights and nMOS for negative weights) is also captured and included in our algorithmic model.

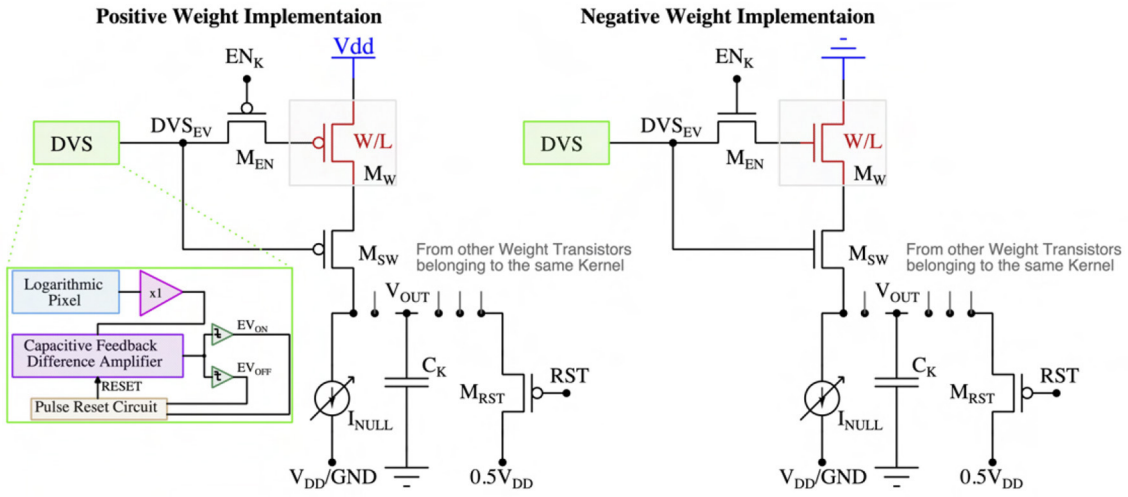

FIGURE 2
Embedded multi-bit positive and negative weight implementation.

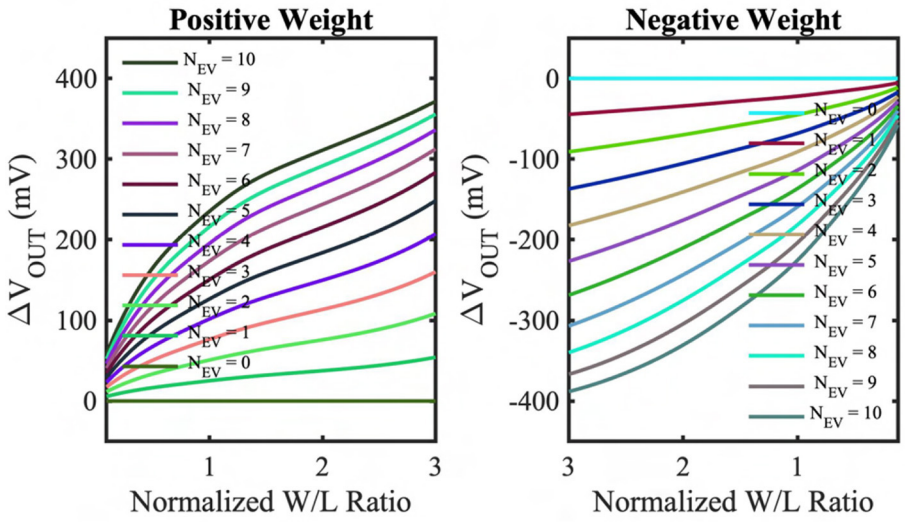

FIGURE 3
Output accumulation voltage change (ΔV_{OUT}) from the reset voltage of the kernel capacitor (C_K) as a function of normalized weight (normalized transistor $\frac{W}{L}$ ratio) and input event spikes simulated on GF 22nm FD-SOI node for positive and negative weights.

2.2. *In-situ* multi-pixel multi-channel convolution operation

In the first spiking CNN layer, we must perform *spatio-temporal* MAC operations across multiple channels *simultaneously* for each kernel. Figure 4 illustrates our proposed neuromorphic-P^{2}M architecture. The left sub-figure represents an array of DVS pixels (each white rectangular box includes multiple DVS pixels arranged in rows and columns) consisting of multiple channels distributed spatially. Each DVS pixel is connected with multiple weight transistors of the analog MAC blocks depending on the number of channels and stride (e.g., each DVS pixel will be connected with four sets of analog MAC blocks for a stride of 2). Each channel performs analog MAC operations asynchronously for a fixed temporal window (the length of each algorithmic time step). For instance, assume the kernel size is 3 × 3, and each kernel has 5 different channels that are represented by the white rectangular boxes in the left sub-figure. The right sub-figure exhibits the zoomed version of the 3 × 3 kernel with 5 different channels. Each channel has a dedicated accumulation capacitor (e.g., C_{Ki}, where i = 1, 2, ... 5) and a local bitline so that charge can accumulate across all the different channels at the same time. Depending on the kernel size, multiple weight transistors (both positive and negative) are connected to its kernel-dedicated accumulation capacitor using the local bitline of each channel. In this example, 18 weight transistors (kernel size = 3 × 3 and for the ON and OFF channels of the

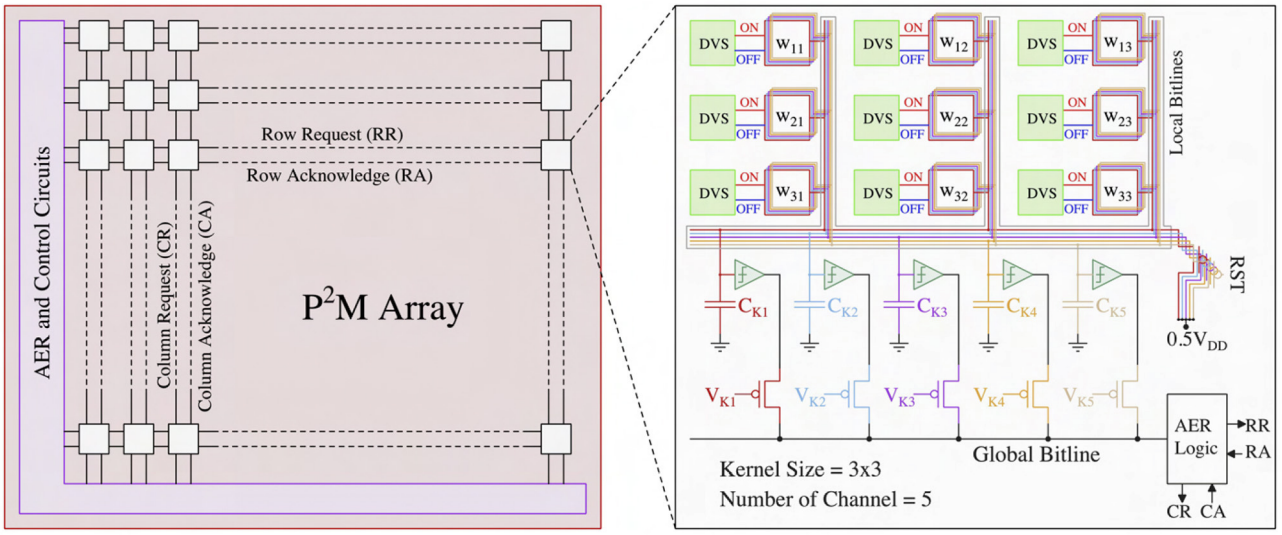

FIGURE 4
Neuromorphic-P^{2}M array block diagram with peripheral control circuits and multi-channel configuration of our proposed neuromorphic-P^{2}M architecture.

DVS pixels) are connected with a single kernel capacitor. The per-channel accumulation capacitor and local bitline shared among the kernel's weight transistors enable simultaneously and massively parallel spatio-temporal MAC operations across different channels. The multiplication results (fixed amount of charge transfer to kernel capacitor from V_{DD} or from kernel capacitor to GND as a function of positive or negative weight depending on the weight values) accumulate on the kernel capacitor for a fixed temporal window (length of each algorithmic time step). These analog MAC operations are asynchronous and parallel across all the kernels for all the input feature maps (DVS pixels) throughout the sensor array. Finally, a thresholding circuit compares the final accumulated voltage on each channel's capacitor with a reference voltage to generate the output activation spike. Output activations from different channels are multiplexed (controlled by V_{K1}, V_{K2}, etc.) to communicate with the AER read circuits at the periphery (left sub-figure) through the kernel-level AER logic block (right sub-figure). The row request (RA) and row acknowledge (RA) signals are shared along the rows, and the column request (CR) and column acknowledge (CA) signals are shared along the columns. After the read operation (described in Section 2.3), the kernel's accumulation capacitor is reset to $0.5V_{DD}$ by the reset transistor (M_{RST}) shown in Figure 5. Note that the reset operation implies no propagation of the voltage accumulated on the kernel's capacitor from one time step to subsequent time steps. Thus, the kernel capacitor voltage is unlike the typical representation of the membrane potential found in the literature (Datta et al., 2021, 2022a,b), which is conserved across time steps. Taking into cognizance the above behavior, for the first layer of the network, we ensure our algorithmic framework includes thresholding and reset operation across time steps, thus faithfully representing the circuit behavior in algorithmic simulations.

The frequency of the reset operation is based on the amount of time the capacitor can hold the charge without significant leakage. To minimize the capacitor leakage, we use high-V_T weight transistors, a switching transistor (M_{SW}) to disconnect the kernel's capacitor from the weight transistors, and kernel-dependent nullifying current source (I_{NULL}) (shown in Figure 2). According to our HSpice simulations, in the worst-case scenario (all weights are maximum in the kernel, which is very unlikely in the neural network model), the voltage on the accumulation capacitor deviates due to leakage from its ideal value by a mere 22 mV over a significantly longer duration of time (e.g., 1 ms). Based on the reset frequency, the length of each algorithmic time step of our neural network model has been set to 1 ms for the first layer.

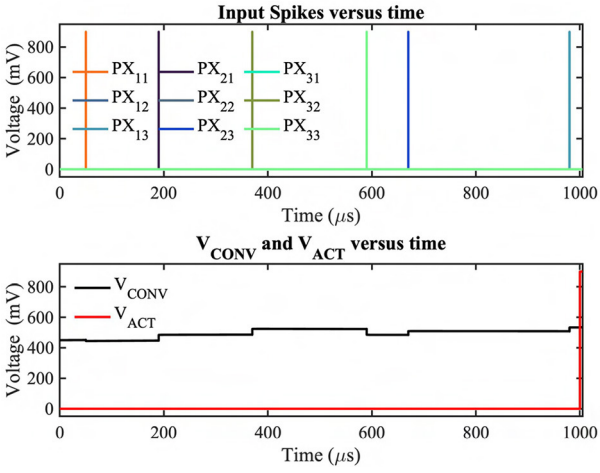

FIGURE 5
A random convolution operation with output activation spike simulated on GF 22nm FD-SOI node.

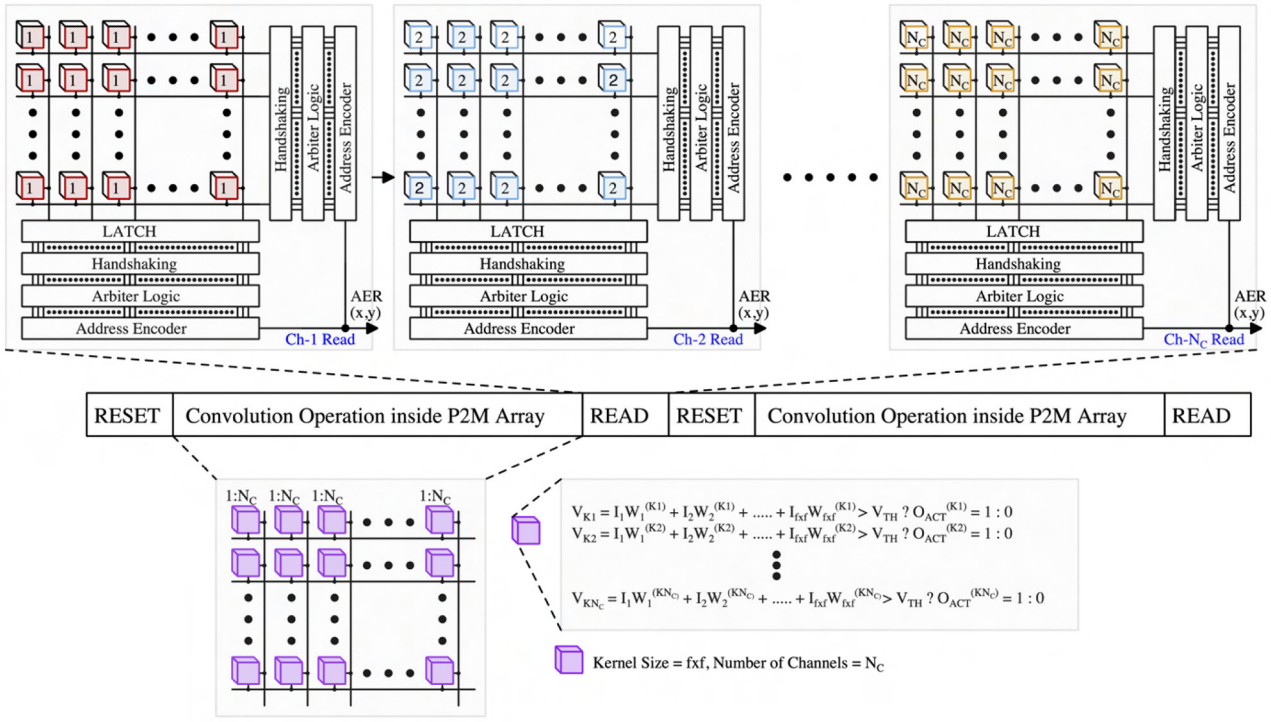

FIGURE 6
AER read scheme of our proposed neuromorphic-P^{2}M architecture.

Figure 5 illustrates an asynchronous convolution and output activation spike generation example of our proposed neuromorphic-P^{2}M using the GF22nm FD-SOI technology node considering random inputs and weights. For this simulation, a kernel size of 3 × 3 has been considered. The weights, the instant of the events, and the number of events per DVS pixel are generated randomly. For the test HSpice simulation, 1 ms simulation time has been considered according to our algorithmic framework; hence, all the output events from DVS pixels within this time period will be multiplied with their weights and accumulated on the Kernel's accumulation capacitor before being compared with a fixed threshold voltage. The top subplot exhibits that the DVS pixels (e.g., PX_{11}, PX_{21}, etc.) are generating the event spikes at different time instants. PX_{13}, PX_{21}, PX_{23}, PX_{32} are connected with positive weights, whereas the other pixels are connected with negative weights. It may also be noted that a few pixels (e.g., PX_{12}, PX_{22}, PX_{31}) do not generate any event during this time frame. This test simulation also considers these no-event generation scenarios to mimic the actual dataset sparsity. From the bottom subplot, it can be observed that the convolution output (V_{CONV}) of our analog MAC circuit is updating (charging or discharging) for each input event spike. When the weight is positive (negative), the accumulation voltage steps up (down) depending on the weight value. Finally, after the fixed temporal window, the convolution output has been compared with the threshold voltage. If the convolution output is higher than the threshold voltage, the comparator will generate an output activation spike (V_{ACT}) for the next layer for each kernel.

2.3. P^{2}M address-event representation (AER) read operation

In this sub-section, we propose modifications to the standard AER scheme in a manner so that it can be compatible with the presented asynchronous processing in-pixel computations. We are utilizing the asynchronous AER read-out scheme (Boahen, 2004) to read the output activations from the first convolution layer (mapped onto the DVS pixels using our proposed neuromorphic-P^{2}M paradigm). The representative read scheme is illustrated in Figure 6. Our P^{2}M architecture can support multiple numbers of channels (e.g., N_C) as required by the spiking CNN model. The outputs of the channels (thresholded output activation spikes) are read sequentially throughout the P^{2}M array in an asynchronous manner. At a time, one channel is being asserted of the P^{2}M array by activating V_{Ki} sequentially, where i = 1, 2, ... N_C (shown in Figure 4). Kernel-level AER logic block shared among different channels for each spatial feature map generates the row, and column request signals whenever an output activation spike exists in the kernel. For AER reading, row-parallel techniques can be used where it latches all the events generated in a single row and read them sequentially (Boahen, 2004). The peripheral address encoders (row and column encoders) of the AER read circuits output the x and y addresses of the output activation in parallel. Moreover, while performing the read operation, we can also pipeline the next reset and convolution phases without waiting for the read phase to be completed by adding a transistor between the kernel capacitor and the comparator. The comparator output can be stored on the

dynamic node for a short period of time, or even we can use a small holding capacitor to hold the output activation for a sufficient amount of time considering the read operation. As the output activations are sparse and AER read can be completed within a few μs windows, we can also utilize our architecture to perform the convolution and read phase in parallel. Besides, performing the in-pixel convolution operation reduces the output activation map size as a function of the kernel size and number of strides. In addition, we also do not need to send an extra bit to define the polarity of the event (ON or OFF-event), similar to the base DVS systems. As a result, the required number of address bits that need to be communicated off-chip has been reduced from the base DVS system. Hence, our P^{2}M architecture maintains the energy benefit of a sparse system due to utilizing the AER read scheme along with lower off-chip communication energy cost due to generating fewer address bits per output activation.

2.4. Process integration and area consideration

Figure 7 exhibits the representative illustration of a heterogeneously integrated system featuring our proposed neuromorphic-P^{2}M paradigm. Our proposed system can be divided into two key dies, i) a backside illuminated CMOS image sensor (BI-CIS) consisting of the DVS pixels and biasing circuitry, and ii) a die containing multi-bit multi-channel weight transistors, accumulation capacitors, comparators, and AER read circuits. Figure 4 shows that for each spatial feature (DVS pixels), the algorithm requires multiple channels that incur higher area due to multiple weight transistors and one accumulation capacitor per channel. However, due to the advantages of heterogeneous integration, our bottom die can be fabricated on an advanced technology node compared to the top die (BI-CIS). Hence, multiple channels in the bottom die can be accommodated and aligned with the top die without any area overhead while maintaining the neural network model accuracy. It may be noted that typical DVS pixels are larger due to the inclusion of a capacitive feedback difference amplifier. The overall system can be fabricated by a wafer-to-wafer bonding process using pixel-level hybrid Cu2Cu interconnects (Kagawa et al., 2016; Miura et al., 2019; Seo et al., 2021). Each DVS pixel has two Cu2Cu interconnects for its ON and OFF-channel, respectively. Considering the DVS pixel area of 40 μm × 40 μm (Lichtsteiner et al., 2008) for 128 × 128 sensor array, Cu2Cu hybrid bonding pitch of 1 μm (Kagawa et al., 2020) and the analog convolution elements (weight transistors, comparators, accumulation capacitors) area in GF22nm FD-SOI node, our neuromorphic-P^{2}M architecture can support a maximum of 128 and 32 channels with a kernel size of 3 × 3 for stride 2 and 1, respectively. However, 32 channels with stride 2 have been utilized in our algorithmic framework. Such kernel-parallel MAC structure allows us to enable *in-situ* convolution operation without needing weight transfer from a different physical location; thus, this method does not lead to any data bandwidth or energy bottleneck.

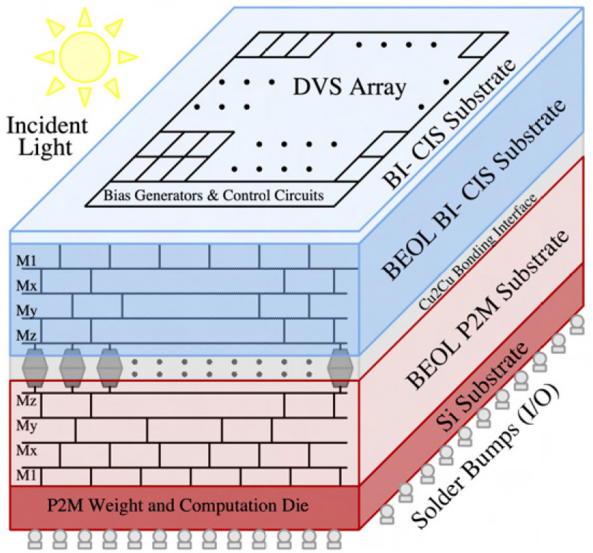

FIGURE 7
Representative illustration of a heterogeneously integrated system featuring neuromorphic-P^{2}M paradigm utilizing Cu2Cu bonding.

3. P^{2}M-constrained algorithm-hardware co-design

This section presents our algorithmic framework implementation guided by our proposed neuromorphic-P^{2}M architecture. The in-pixel charge-based analog convolution generates non-ideal non-linear convolution; in addition, process variation yields a deviation of the convolution result from the ideal output. Moreover, leakage poses constraints on the maximum length of each algorithmic time step, and the area limits the number of channels utilized per each spatial feature map. The hardware-algorithm co-design framework of our proposed neuromorphic-P^{2}M approach has been illustrated in Figure 8. More details on including non-idealities, process variation, leakage, and area effects in our algorithmic framework are given in the following subsections.

3.1. Custom convolution for the first layer modeling circuit non-linearity and process variation

From an algorithmic perspective, the first layer of a spiking CNN is a linear convolution layer followed by a non-linear activation unit. In our neuromorphic-P^{2}M paradigm, we have implemented the weights utilizing voltage accumulation through appropriately sized transistors that are inherently non-linear. As a result, any analog convolution circuit built on transistor devices will exhibit non-ideal non-linear behavior. Hence, to suppress the non-linearity, we have tuned our weights (transistor's geometry) in

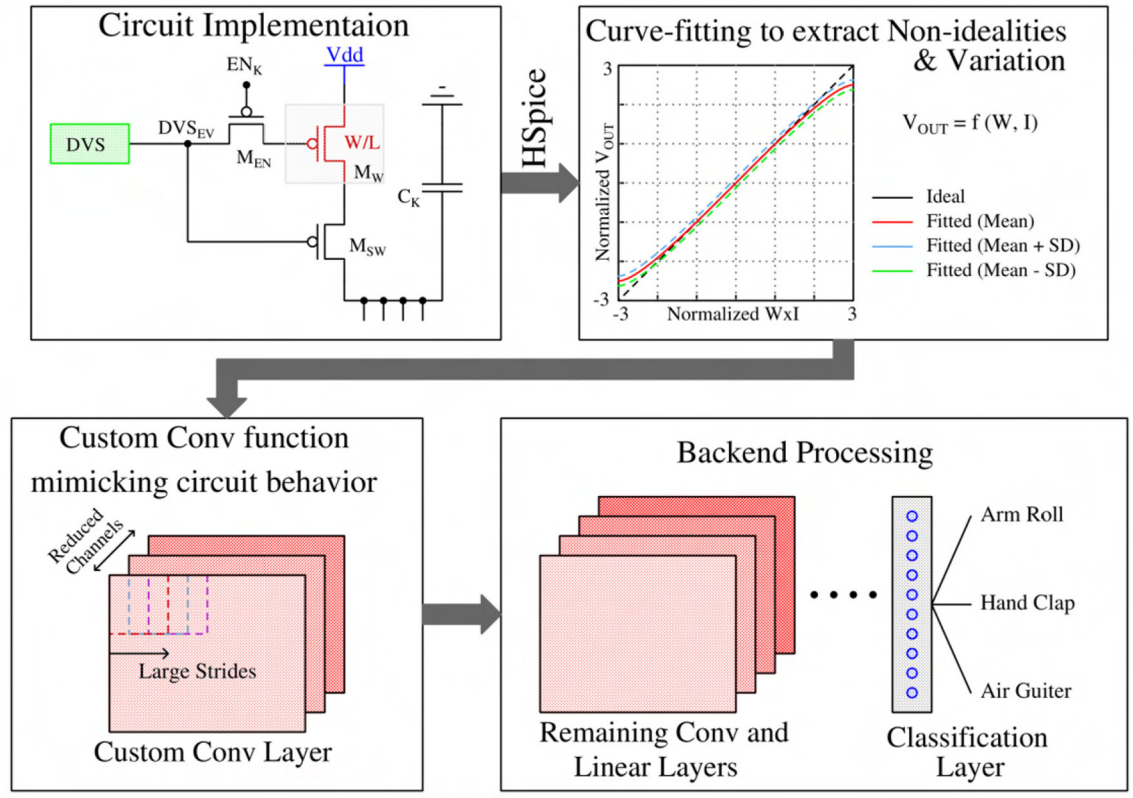

FIGURE 8
Hardware-algorithm co-design framework to enable our proposed neuromorphic-P^{2}M approach.

a non-linear manner in such a way that the output accumulation voltage steps can increase or decrease linearly for positive and negative weights, respectively. However, the nonlinearity is also a function of the drain-to-source voltage of the weight transistors. In our scheme, we are charging or discharging the kernel's capacitor during the computation phase. Depending on the weight values, the charging and discharging current are functionally dependent on the drain-to-source voltage. Hence, when the accumulation voltage on the V_{OUT} node (shown in Figure 2) gets larger (smaller) for the positive (negative) weights, the transistor enters into the triode region; hence, the charging or discharging current reduces. Besides, the same positive and negative weight values cannot ensure the same change in voltage accumulation due to device asymmetry (pMOS for positive weight implementation and nMOS for negative weight implementation). Furthermore, due to process variation, the transistor's geometry cannot be fabricated precisely; hence, the convolution output current can also vary due to process variation. Considering all these non-linear non-ideal behaviors and process variations, we extensively simulated our proposed P^{2}M paradigm for a wide range of input spikes and weights combinations considering leakage and around 3-sigma variation using GF22nm FD-SOI technology node. Figure 9 illustrates the resulting HSpice results with a standard deviation bar, i.e., the normalized convolution output voltages per pixel corresponding to a range of weights and input number of spikes have been modeled using a behavioral curve-fitting function. Note, for the scatter plot, we have used 100 μs temporal window for the convolution phase to save the total circuit simulation time as we have to run 1,000 Monte-Carlo simulations for each combination of weights and the number of input spikes. In our algorithmic framework, a random Gaussian sample value has been generated between the mean $\pm$ standard deviation for each particular normalized weight times input event value to capture the effects of the process variation. For the fixed simulation time for the event stream, in each Kernel, the accumulation output voltage per pixel is calculated first, then added to the other pixel's accumulation voltage inside the kernel to calculate the final output. The algorithmic framework was then used to optimize the spiking CNN training for the event-driven neuromorphic datasets. Besides the above-mentioned non-ideality and variation effects, thermal noise and temperature variation may affect the inference performance. The thermal noise of the circuit can also be modeled as zero-mean Gaussian distribution (Gow et al., 2007). Hence, this can be incorporated by adding an appropriate standard deviation with the mean and standard deviation for the process variation in our framework. Moreover, temperature variation can increase or decrease the step size on each kernel's accumulation capacitor. Large deviation (higher than 3-sigma of the process variation) from the nominal step size due to temperature can affect the classification accuracy.

To validate our Hspice simulations generated curve-fitting function's prediction accuracy, we have tested 1,000 random cases. In these test cases, we have used a kernel size of 3 $\times$ 3, where

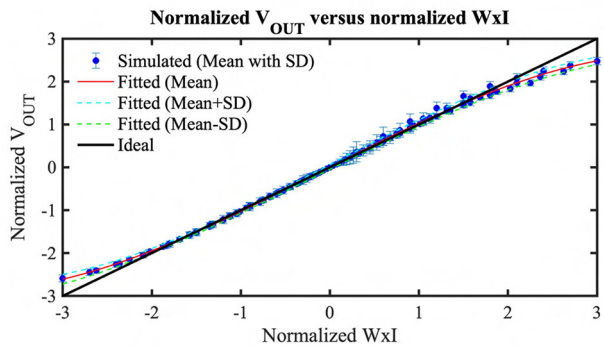

FIGURE 9
A scatter plot with standard deviation comparing the pixel output voltage to ideal multiplication value of Weights × Input activation (Normalized W × I).

the weight values are generated randomly. Moreover, the number of input event spikes and time instants for the input spikes are also randomly generated. Note these random tests utilize 100 μ s of simulation time for the convolution phase to reduce the total simulation time. As mentioned earlier, utilizing kernel-dependent nullifying current source, high-V_T weight transistors and a switch to disconnect the kernel's capacitor exhibits a maximum of 22 mV error in the worst-case scenario. Hence, random HSpice tests ignoring a long time (1 ms of simulation time length for each event stream of our neural network model) will not incur any significant accuracy issues for these 1,000 random tests. Among 1,000 random tests, only 100 test results (for clear visibility) are shown in Figure 10. The figure shows the curve-fitted mean and mean± standard deviation predictions of our proposed analog MAC operations with HSpice-generated simulation results. We have used a 3rd order single variable (normalized weight times input event spikes) polynomial to generate the curve fitting functions (mean, mean ± standard deviation) considering 0.55% mean RMSE of our analog MAC to minimize the computation complexity in our algorithmic framework while maintaining high accuracy. It can clearly be seen that the predicted mean output follows the Hspice results closely, and the HSpice outputs fall between the mean± standard deviation value.

3.2. Circuit-algorithm co-optimization of spiking CNN backbone subject to P^{2}M constraints

In our proposed neuromorphic-P^{2}M architecture, we have utilized a kernel-dedicated capacitor to enable instantaneous and massively parallel spatio-temporal convolution operation across different channels. We need a kernel-dedicated capacitor to preserve the temporal information of input DVS spikes across different channels simultaneously. Moreover, there is a direct trade-off between the acceptable leakage and capacitance value (a large capacitor incurs a large area; however, it results in lower leakage). Almost 47% of the area in our P^{2}M array is occupied by the capacitors. Hence, we have reduced the number of channels in our spiking CNN models compared to the baseline neural architecture not to incur any area overhead while preserving the model accuracy. In addition, the leakage also limits the length of each algorithmic time step in our algorithmic framework. We have also reduced the time length in our neural network model to minimize the kernel-dependent leakage error of our custom first convolution layer. Moreover, to reduce the amount of data transfer between the P^{2}M architecture and the backend hardware processing of the remaining spiking CNN layers, we have avoided the max pooling layer and instead used a stride of 2 in the P^{2}M convolutional layer. Lastly, we incorporate the Monte Carlo variations in the proposed non-linear custom convolutional layer explained above in our algorithmic framework. In particular, we have estimated the mean and standard deviation of the output of the custom convolutional layer from extensive circuit simulations. We then train our spiking CNN with the addition of the standard deviation as noise to the mean output of the convolutional layer. This noise addition during training is crucial to increase the robustness of our spiking CNN models, as otherwise, our models would incur a drastic drop in test accuracy.

In this work, we have evaluated our P^{2}M paradigm on complex neuromorphic datasets where each event is at least a few seconds long. Hence, with a timestep length of 1 ms, we will require more than thousands of total time steps to train our SNNs with these neuromorphic datasets. This is impractical (it would require more than a year to train one SNN model on the DVS Gesture dataset) in modern GPUs typically used for training SNNs. To mitigate this problem, we only employed a small timestep length (1 ms) in our first P^{2}M-implemented layer where the weights are kept frozen while the remaining layers implemented outside the sensor are trained with a large timestep length that leads to a small number of total time steps. The weights in the P^{2}M-implemented layer are obtained from a baseline SNN where all the layers have the same large timestep length. Thus, the length of the timestep impacts the trainability of the SNNs. It also affects the temporal information injected into the SNN, i.e., as the length of the timestep reduces, the SNN can extract more fine-grained temporal features, which can potentially improve the inference performance at the cost of reduced trainability. We expect to reduce energy consumption by increasing the time step length as that would inject a smaller number of spikes into the network (a constant large incoming synaptic input can emit more spikes if the number of time steps is increased, i.e., the time step length is decreased).

4. Experimental results

4.1. Benchmarking dataset and model

This article focuses on the potential use of P^{2}M for event-driven neuromorphic tasks where the goal is to classify each video sample captured by the DVS cameras. In particular, we evaluate our P^{2}M approach on two large-scale popular neuromorphic benchmarking datasets.

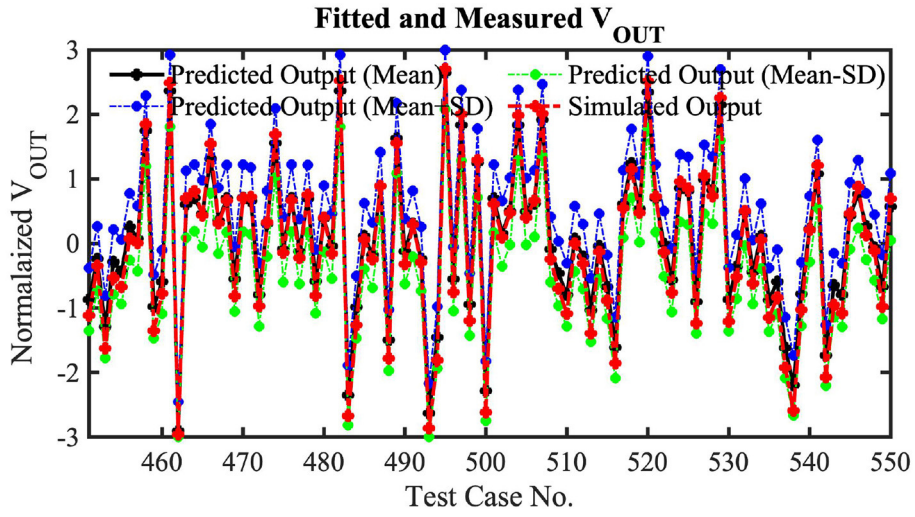

FIGURE 10
One hundred random HSpice simulation results for 3 × 3 Kernel benchmarking with the fitted equations.

4.1.1. DVS128-gesture

The IBM DVS128-Gesture (Amir et al., 2017) is a neuromorphic gesture recognition dataset with a temporal resolution in μs range and a spatial resolution of 128×128. It consists of 11 gestures (1,000 samples each), such as hand clap, arm roll, etc., recorded from 29 individuals under three illumination conditions, and each gesture has an average duration of 6 s. To the best of our knowledge, it is the most challenging open-source neuromorphic dataset with the most precise temporal information.

4.1.2. NMNIST

The neuromorphic MNIST (Orchard et al., 2015) dataset is a converted dataset from MNIST. It consists of 50K training images and 10K validation images. We preprocess it in the same way as in N-Caltech 101. We resize all our images to 34×34.

For these datasets, we apply a 9:1 train-valid split. We use the Spikingjelly package (Fang et al., 2020) to process the data and integrate them into a fixed time interval of 1 ms based on the kernel's capacitor retention time supported by our neuromorphic-P^{2}M circuit. However, such a small integration time would lead to a large number of time steps for the neuromorphic datasets considered in this work whose input samples are at least a few seconds long. This would significantly exacerbate the training complexity. To mitigate this concern, we first pre-train a spiking CNN model with a large integration time in the order of seconds (i.e., with a small number of time steps) without any P^{2}M circuit constraints. We then decrease the integration time of the first spiking convolutional layer for P^{2}M implementation and integrate the spikes in the second interval such that the network from the second layer processes the input with only a few time steps. We fine-tune this network from the second layer while freezing the first layer since training the first layer significantly increases the memory complexity due to a large number of time steps. This is because the gradients of the first layer need to be unrolled across all the time steps.

We use four convolutional layers, followed by two linear layers at the end with 512 and 10 neurons, respectively. Each convolutional layer is followed by a batch normalization layer, spiking LIF layer, and max pooling layer.

4.2. Classification accuracy

We evaluated the performance of the baseline and P^{2}M custom spiking CNN models on the two datasets illustrated above in Table 1. Note that all these models are trained from scratch. As we can see, the custom convolution model does not incur any significant drop in accuracy for any of the two datasets. However, removing the state variable, i.e., the membrane potential in the first layer, leads to an average ∼ 5% drop in test accuracy. This might be because of the loss in the temporal information of the input spike integration from the DVS camera. Additional P^{2}M constraints, such as less number of channels and increased strides in the first convolutional layer (see Section 3.2), hardly incur any additional drop in accuracy. Overall, our P^{2}M-constrained models lead to an average 5.2% drop in test accuracy across the two datasets.

4.3. Analysis of energy consumption

We develop a circuit-algorithm co-simulation framework to characterize the energy consumption of our baseline and P^{2}M-implemented spiking CNN models for neuromorphic datasets. Note that we do not evaluate the latency of our models since that would depend heavily on the underlying hardware architecture and data flow of the backend hardware (i.e., the hardware processing the remaining layers of the CNN, excluding the first layer that is processed using our P^{2}M paradigm). The frontend energy ($E_{frontend}$) is comprised of sensor energy (E_{sens}) and communication energy (E_{com}), while the backend energy

TABLE 1 Comparison of the test accuracy of our P^{2}M enabled spiking CNN models with the baseline spiking CNN counterparts, where "MP" denotes membrane potential, "Custom conv." denotes the incorporation of the non-ideal model to the ML algorithmic framework, and "Reduced dimensionality" denotes the reduction in the number of channels in the first convolutional layer.

Dataset	MP 1st layer	Custom conv.	Reduced dimensionality	Accuracy (%)
DVS128-Gesture	✓	×	×	93.40
DVS128-Gesture	×	×	×	88.78
DVS128-Gesture	×	✓	×	88.54
DVS128-Gesture	×	✓	✓	88.36
NMNIST	✓	×	×	98.10
NMNIST	×	×	×	93.68
NMNIST	×	✓	×	93.44
NMNIST	×	✓	✓	93.12

TABLE 2 Energy estimation for different hardware components.

Model type	Sensing energy (mJ) (E_{sens})	Comm energy (pJ/bit) ($e_{comm} = e_{sens-to-tx} + e_{tx}$)	MAC energy (pJ) (e_{mac})	MAdds energy (pJ) (e_{ac})
P^{2}M (ours)	26.588	4.1	1.568	0.03
Baseline	26.032	4.1	1.568	0.03

The energy values are measured for designs in 22 nm CMOS technology. Note, the sensing energy (E_{sens}) of our model includes the convolution energy for P^{2}M as the convolution is performed as a part of the sensing operation. The communication energy (e_{comm}) includes both the energy consumption of sending the address bits from the sensor to the transmitter ($e_{sens-to-tx}$) and wireless transmitter energy (e_{tx}). For e_{mac} and e_{ac}, we convert the corresponding value in 45 nm to that of 22 nm by following standard scaling strategy (Stillmaker and Baas, 2017).

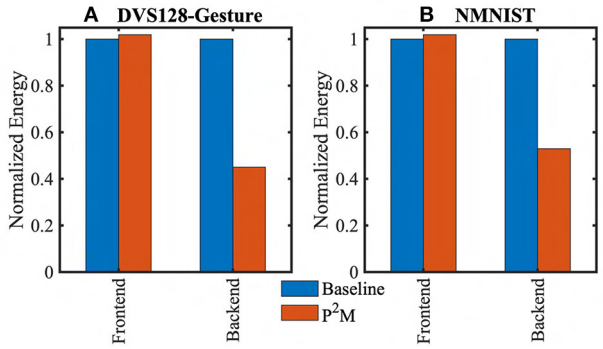

FIGURE 11
Comparison of the energy consumption between baseline and P^{2}M implementations of spiking CNNs to process neuromorphic images from **(A)** DVS128-Gesture, and **(B)** NMNIST datasets.

($E_{backend}$) to process the SNN layers (excluding the first layer for the P^{2}M implementation) is primarily composed of the accumulation operations incurred by the spiking convolutional layers (E_{ac}) and the parameter read (E_{read}) costs. Assuming T denotes the total number of time steps and s denotes the sparsity. The energy components can be approximated as follows:

$$E_{frontend} \approx \underbrace{e_{event} * N_{event} + E_{bias}}_{E_{sens}} + \underbrace{(e_{sens-to-tx} + e_{tx}) * N_{event}}_{E_{com}} \quad (1)$$

$$E_{backend} \approx \underbrace{e_{ac} * N_{ac} * s * T}_{E_{ac}} + \underbrace{e_{read} * N_{read}}_{E_{read}} \quad (2)$$

Here, e_{event} represents per-pixel sensing energy, N_{event} denotes the number of events communicated from the sensor to the backend, and E_{bias} is the biasing energy for the DVS pixel array considering the dataset duration. In addition, $e_{sens-to-tx}$ is the communication energy to send the address bits from the sensor node to the transmitter, and e_{tx} is the wireless transmission energy to the backend. Note that the first convolutional layer of the SNN in the baseline implementation requires MAC operations, and hence, we need to replace e_{ac} with the MAC energy e_{mac} and use s=1. For a spiking convolutional layer that takes an input $\mathbf{I} \in R^{h_i \times w_i \times c_i}$ and weight tensor $\boldsymbol{\theta} \in R^{k \times k \times c_i \times c_o}$ to produce output $\mathbf{O} \in R^{h_o \times w_o \times c_o}$, the N_{ac} (Datta et al., 2021, 2022b; Kundu et al., 2021a,b) and N_{read} can be computed as

$$N_{ac} = h_o * w_o * k^2 * c_i * c_o \quad (3)$$

$$N_{read} = k^2 * c_i * c_o \quad (4)$$

The energy values we have used to evaluate $E_{frontend}$ and $E_{backend}$ are presented in Table 2. While E_{sens} and $e_{sense-to-tx}$ are obtained from our circuit simulations, e_{tx} is obtained from Lin et al. (2021), and e_{ac} and e_{read} are obtained from Kang et al. (2018). Figure 11 shows the comparison of energy costs for standard vs P^{2}M-implemented spiking CNN models for the DVS

datasets. In particular, P^{2}M can yield a backend energy reduction of up to $\sim$ 2$\times$ with the cost of 2% increase in frontend energy only. This reduction primarily comes from the reduced energy consumption in the backend since we offload the compute of the first convolutional layer of the SNN. This layer consumes more than 50% of the total backend energy since it involves expensive MAC operations (due to event accumulation before convolution computation), which consume $\sim$32$\times$ more energy compared to cheap accumulate operations (Horowitz, 2014) with 32-bit fixed point representation. Thus, the proposed neuromorphic-P^{2}M paradigm enables *in-situ* availability of the weight matrix within the array of DVS pixels (reducing the energy overhead associated with the transfer of weight matrix) while also significantly reducing energy-consumption of MAC operations by utilizing massively parallel non-von-Neumann analog processing-in-pixel.

5. Conclusion

We have proposed and implemented a novel in-pixel-in-memory processing paradigm for neuromorphic event-based sensors in this work. To the best of our knowledge, this is the first proposal to enable massively parallel, energy-efficient non-von-Neumann analog processing-in-pixel for neuromorphic image sensors using novel weight-embedded pixels. Instead of generating event spikes based on the change in contrast of scenes, our proposed solutions can directly send the low-level output features of the convolutional neural network using a modified address event representation scheme. By leveraging advanced 3D integration technology, we can perform *in-situ* massively parallel charge-based analog spatio-temporal convolution across the pixel array. Moreover, we have incorporated the hardware (non-linearity, process variation, leakage) constraints of our analog computing elements as well as area consideration (limiting the maximum number of channels of the first neural network layer) into our algorithmic framework. Our P^{2}M-enabled spiking CNN model yields an accuracy of 88.36% on the IBM DVS128-Gesture dataset and achieved $\sim$ 2$\times$ backed energy reduction compared to the conventional system.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://research.ibm.com/interactive/dvsgesture/, https://www.garrickorchard.com/datasets/n-mnist.

Author contributions

MK proposed the use of P^{2}M for the neuromorphic vision sensor and developed the corresponding circuit simulation framework. GD developed the baseline and P^{2}M-constrained algorithmic framework with the help of ZW. APJ and ARJ proposed the idea of P^{2}M. MK and GD wrote the majority of the paper. ARJ and PAB supervised the research and reviewed the manuscript extensively. All authors reviewed the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was funded in part by the DARPA HR00112190120 award.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., et al. (2017). "A low power, fully event-based gesture recognition system," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7388–7397.

Beltrán, J., Guindel, C., Cortés, I., Barrera, A., Astudillo, A., Urdiales, J., et al. (2020). "Towards autonomous driving: a multi-modal 360° perception proposal," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (IEEE), 1–6.

Boahen, K. A. (2004). A burst-mode word-serial address-event link-I: Transmitter design. *IEEE Trans. Circ. Syst. I Regular Pap.* 51, 1269–1280. doi: 10.1109/TCSI.2004.830703

Bose, L., Dudek, P., Chen, J., Carey, S. J., and Mayol-Cuevas, W. W. (2020). "Fully embedding fast convolutional networks on pixel processor arrays," in *European Conference on Computer Vision* (Springer), 488–503.

Chai, Y. (2020). In-sensor computing for machine vision. *Nature* 579, 32–33 doi: 10.1038/d41586-020-00592-6

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Process. Mag.* 37, 34–49. doi: 10.1109/MSP.2020.2985815

Chen, Z., Zhu, H., Ren, E., Liu, Z., Jia, K., Luo, L., et al. (2019). Processing near sensor architecture in mixed-signal domain with cmos image sensor of convolutional-kernel-readout method. *IEEE Trans. Circ. Syst. I Reg. Pap.* 67, 389–400. doi: 10.1109/TCSI.2019.2937227

Cheng, W., Luo, H., Yang, W., Yu, L., and Li, W. (2020). Structure-aware network for lane marker extraction with dynamic vision sensor. *arXiv preprint arXiv:2008.06204*. doi: 10.48550/arXiv.2008.06204

Datta, G., and Beerel, P. A. (2022). "Can deep neural networks be converted to ultra low-latency spiking neural networks?" in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 718–723.

Datta, G., Deng, H., Aviles, R., and Beerel, P. A. (2022a). Towards energy-efficient, low-latency and accurate spiking LSTMs. *arXiv preprint arXiv:2110.05929*. doi: 10.48550/arXiv.2210.12613

Datta, G., Kundu, S., and Beerel, P. A. (2021). "Training energy-efficient deep spiking neural networks with single-spike hybrid input encoding," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Datta, G., Kundu, S., Jaiswal, A. R., and Beerel, P. A. (2022b). ACE-SNN: algorithm-hardware co-design of energy-efficient & low-latency deep spiking neural networks for 3D image recognition. *Front. Neurosci.* 16, 815258. doi: 10.3389/fnins.2022.815258

Datta, G., Kundu, S., Yin, Z., Lakkireddy, R. T., Mathai, J., Jacob, A. P., et al. (2022c). A processing-in-pixel-in-memory paradigm for resource-constrained tinyml applications. *Sci. Rep.* 12:14396. doi: 10.1038/s41598-022-17934-1

Datta, G., Kundu, S., Yin, Z., Mathai, J., Liu, Z., Wang, Z., et al. (2022d). P2M-DeTrack: processing-in-Pixel-in-Memory for energy-efficient and real-time multi-object detection and tracking. *arXiv preprint arXiv:2205.14285*. doi: 10.1109/VLSI-SoC54400.2022.9939582

Datta, G., Yin, Z., Jacob, A., Jaiswal, A. R., and Beerel, P. A. (2022e). Toward efficient hyperspectral image processing inside camera pixels. *arXiv preprint arXiv:2203.05696*. doi: 10.48550/arXiv.2203.05696

Eki, R., Yamada, S., Ozawa, H., Kai, H., Okuike, K., Gowtham, H., et al. (2021). "9.6 a 1/2.3 inch 12.3 mpixel with on-chip 4.97 tops/w CNN processor back-illuminated stacked CMOS image sensor," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE), 154–156.

Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., et al. (2020). *Spikingjelly*. Available online at: https://github.com/fangwei123456/spikingjelly

Goel, A., Tung, C., Lu, Y.-H., and Thiruvathukal, G. K. (2020). "A survey of methods for low-power deep learning and computer vision," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (IEEE), 1–6.

Gow, R. D., Renshaw, D., Findlater, K., Grant, L., McLeod, S. J., Hart, J., et al. (2007). A comprehensive tool for modeling cmos image-sensor-noise performance. *IEEE Trans. Electron Dev.* 54, 1321–1329. doi: 10.1109/TED.2007.896718

Horowitz, M. (2014). "Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14.

Hsu, T.-H., Chen, G.-C., Chen, Y.-R., Lo, C.-C., Liu, R.-S., Chang, M.-F., et al. (2022). "A 0.8 v intelligent vision sensor with tiny convolutional neural network and programmable weights using mixed-mode processing-in-sensor technique for image classification," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE), 1–3.

Hsu, T.-H., Chen, Y.-R., Liu, R.-S., Lo, C.-C., Tang, K.-T., Chang, M.-F., et al. (2020). A 0.5-v real-time computational cmos image sensor with programmable kernel for feature extraction. *IEEE J. Solid State Circ.* 56, 1588–1596. doi: 10.1109/JSSC.2020.3034192

Jaiswal, A., and Jacob, A. P. (2021). *Integrated Pixel and Two-Terminal Non-Volatile Memory Cell and an Array of Cells for Deep In-sensor, In-Memory Computing*. US Patent 11195580.

Jiao, L., Zhang, R., Liu, F., Yang, S., Hou, B., Li, L., et al. (2022). New generation deep learning for video object detection: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 3195–3215. doi: 10.1109/TNNLS.2021.3053249

Jogin, M., Madhulika, M., Divya, G., Meghana, R., and Apoorva, S. (2018). "Feature extraction using convolution neural networks (CNN) and deep learning," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (IEEE), 2319–2323.

Kagawa, Y., Fujii, N., Aoyagi, K., Kobayashi, Y., Nishi, S., Todaka, N., et al. (2016). "Novel stacked CMOS image sensor with advanced cu2cu hybrid bonding," in *2016 IEEE International Electron Devices Meeting (IEDM)* (IEEE), 8–4.

Kagawa, Y., Hashiguchi, H., Kamibayashi, T., Haneda, M., Fujii, N., Furuse, S., et al. (2020). "Impacts of misalignment on 1μm pitch cu-cu hybrid bonding," in *2020 IEEE International Interconnect Technology Conference (IITC)* (IEEE), 148–150.

Kang, M., Lim, S., Gonugondla, S., and Shanbhag, N. R. (2018). An in-memory VLSI architecture for convolutional neural networks. *IEEE J. Emerg. Select. Top. Circ. Syst.* 8, 494–505. doi: 10.1109/JETCAS.2018.2829522

Kundu, S., Datta, G., Pedram, M., and Beerel, P. A. (2021a). "Spike-thrift: towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3953–3962.

Kundu, S., Datta, G., Pedram, M., and Beerel, P. A. (2021b). Towards low-latency energy-efficient deep snns via attention-guided compression. *arXiv preprint arXiv:2107.12445*. doi: 10.48550/arXiv.2107.12445

Lefebvre, M., Moreau, L., Dekimpe, R., and Bol, D. (2021). "7.7 a 0.2-to-3.6 tops/w programmable convolutional imager soc with in-sensor current-domain ternary-weighted mac operations for feature extraction and region-of-interest detection," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE), 118–120.

Leñero-Bardallo, J. A., Serrano-Gotarredona, T., and Linares-Barranco, B. (2011). A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE J. Solid State Circ.* 46, 1443–1455. doi: 10.1109/JSSC.2011.2118490

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128x128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid State Circ.* 43, 566–576. doi: 10.1109/JSSC.2007.914337

LiKamWa, R., Hou, Y., Gao, J., Polansky, M., and Zhong, L. (2016). Redeye: analog convnet image sensor architecture for continuous mobile vision. *ACM SIGARCH Comput. Arch. News* 44, 255–266. doi: 10.1145/3007787.3001164

LiKamWa, R., Wang, Z., Carroll, A., Lin, F. X., and Zhong, L. (2014). "Draining our glass: an energy and heat characterization of google glass," in *Proceedings of 5th Asia-Pacific Workshop on Systems*, 1–7.

Lin, J., and Boahen, K. (2009). "A delay-insensitive address-event link," in *2009 15th IEEE Symposium on Asynchronous Circuits and Systems* (IEEE), 55–62.

Lin, L., Ahmed, K. A., Salamani, P. S., and Alioto, M. (2021). "Battery-less IoT sensor node with pll-less wifi backscattering communications in a 2.5-μw peak power envelope," in *2021 Symposium on VLSI Circuits* (IEEE), 1–2.

Ma, T., Jia, K., Zhu, X., Qiao, F., Wei, Q., Zhao, H., et al. (2019). "An analog-memoryless near sensor computing architecture for always-on intelligent perception applications," in *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)* (IEEE), 150–155.

Maheepala, M., Joordens, M. A., and Kouzani, A. Z. (2020). Low power processors and image sensors for vision-based iot devices: a review. *IEEE Sensors J.* 21, 1172–1186. doi: 10.1109/JSEN.2020.3015932

Mansour, R. F., Escorcia-Gutierrez, J., Gamarra, M., Villanueva, J. A., and Leal, N. (2021). Intelligent video anomaly detection and classification using faster rcnn with deep reinforcement learning model. *Image Vision Comput.* 112, 104229. doi: 10.1016/j.imavis.2021.104229

Maqueda, A. I., Loquercio, A., Gallego, G., García, N., and Scaramuzza, D. (2018). "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5419–5427.

Miura, T., Sakakibara, M., Takahashi, H., Taura, T., Tatani, K., Oike, Y., et al. (2019). "A 6.9 μm pixel-pitch 3d stacked global shutter cmos image sensor with 3m cu-cu connections," in *2019 International 3D Systems Integration Conference (3DIC)* (IEEE), 1–2.

Nguyen, A., Do, T.-T., Caldwell, D. G., and Tsagarakis, N. G. (2019). "Real-time 6dof pose relocalization for event cameras with stacked spatial LSTM networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Orchard, G., Jayawant, A., Cohen, G. K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* 9, 437. doi: 10.3389/fnins.2015.00437

Pinkham, R., Berkovich, A., and Zhang, Z. (2021). Near-sensor distributed DNN processing for augmented and virtual reality. *IEEE J. Emerg. Select. Top. Circ. Syst.* 11, 663–676. doi: 10.1109/JETCAS.2021.3121259

Seo, M.-W., Chu, M., Jung, H.-Y., Kim, S., Song, J., Lee, J., et al. (2021). "A 2.6 e-rms low-random-noise, 116.2 mw low-power 2-mp global shutter cmos image sensor with pixel-level adc and in-pixel memory," n *2021 Symposium on VLSI Technology* (IEEE), 1–2.

Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.-S., Shin, C., et al. (2017). "4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE), 66–67.

Stillmaker, A., and Baas, B. (2017). Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. *Integration* 58, 74–81. doi: 10.1016/j.vlsi.2017.02.002

Wu, X., Li, W., Hong, D., Tao, R., and Du, Q. (2021). Deep learning for unmanned aerial vehicle-based object detection and tracking: a survey. *IEEE Geosci. Remote Sens. Mag.* 10, 91–124. doi: 10.1109/MGRS.2021.3115137

Xie, J., Zheng, Y., Du, R., Xiong, W., Cao, Y., Ma, Z., et al. (2021). Deep learning-based computer vision for surveillance in its: evaluation of state-of-the-art methods. *IEEE Trans. Vehicular Technol.* 70, 3027–3042. doi: 10.1109/TVT.2021.3065250

Xu, H., Lin, N., Luo, L., Wei, Q., Wang, R., Zhuo, C., et al. (2021). Senputing: an ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing. *IEEE Trans. Circ. Syst. I Reg. Pap.* 69, 232–243. doi: 10.1109/TCSI.2021.3090668

Xu, H., Nazhamaiti, M., Liu, Y., Qiao, F., Wei, Q., Liu, X., et al. (2020). "Utilizing direct photocurrent computation and 2d kernel scheduling to improve in-sensor-processing efficiency," in *2020 57th ACM/IEEE Design Automation Conference (DAC)* (IEEE), 1–6.

Zhou, F., and Chai, Y. (2020). Near-sensor and in-sensor computing. *Nat. Electron.* 3, 664–671. doi: 10.1038/s41928-020-00501-9

Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2018). EV-flowNet: self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*. doi: 10.48550/arXiv.1802.06898