# The CNN vs. SNN Event-camera Dichotomy and Perspectives For Event-Graph Neural Networks

Thomas Dalgaty*, Thomas Mesquida*, Damien Joubert†, Amos Sironi†, Cyrille Soubeyrat†,
Pascal Vivet*, Christoph Posch†
*Université Grenoble-Alpes, CEA-List, Grenoble, France
†Prophesee, Paris, France

***Abstract*—Since neuromorphic event-based pixels and cameras were first proposed, the technology has greatly advanced such that there now exists several industrial sensors, processors and toolchains. This has also paved the way for a blossoming new branch of AI dedicated to processing the event-based data these sensors generate. However, there is still much debate about which of these approaches can best harness the inherent sparsity, low-latency and fine spatiotemporal structure of event-data to obtain better performance and do so using the least time and energy. The latter is of particular importance since these algorithms will typically be employed near or inside of the sensor at the edge where the power supply may be heavily constrained. The two predominant methods to process visual events - convolutional and spiking neural networks - are fundamentally opposed in principle. The former converts events into static 2D frames such that they are compatible with 2D convolutions, while the latter computes in an event-driven fashion naturally compatible with the raw data. We review this dichotomy by studying recent algorithmic and hardware advances of both approaches. We conclude with a perspective on an emerging alternative approach whereby events are transformed into a graph data structure and thereafter processed using techniques from the domain of graph neural networks. Despite promising early results, algorithmic and hardware innovations are required before this approach can be applied close or within the Event-based sensor.**

***Index Terms*—Event-camera, Edge AI, neuromorphic computing**

## I. Introduction

Since the late 1980's several pioneering works have applied the analogue properties of transistors to mimic mechanisms such as transient gain adaptation, filtering and lateral gating studied in the early stages of mammalian and insect visual systems [1]–[3]. A similar line of work has also explored active pixel sensing concepts where pixels integrate extra functionality to locally calculate temporal pixel intensity differences [4]. This principle was used to generate a quantity referred to as an *event* [5], signalling a local relative luminosity change at a pixel. Naturally, these two lines of research were combined [6] and have since given rise to what we know today as event-cameras (also often referred to as dynamic vision sensors and silicon retinas). Relative to frame-based imaging, whereby pixel arrays record light intensity periodically, event-based cameras produce a sparse stream of events - each comprising an XY pixel address, a timestamp and a polarity - generated by contrast features moving across the field of view of a pixel. Crucially, this permits event-cameras to capture an unprecedentedly fine spatiotemporal structure of motion that is lost in-between traditional static frames.

This work is partly funded thanks to the French national program "Programme d'Investissements d'Avenir, IRT Nanoelec" ANR-10-AIRT-05

Once generated, events are typically communicated from the camera, using a time-multiplexed protocol called Address-Event Representation (AER) [7], to another system that will then use this information in a concrete application. However, what this system should be and what algorithm will be compiled onto are still very much open questions. This is in large part owed to the fact that the format of event-data differs significantly from - as well as offers unique opportunities relative to - the static frames that computer vision algorithms and hardware have evolved alongside. Owing to their low-power operation, ranging from hundreds of microwatts to some tens of millwatts, data-driven massively compressed output (relative to frames), and their high temporal resolution (i.e., low-latency operation), event-cameras have significant potential in edge computer vision and artificial intelligence applications based on fast moving and highly dynamic visual scenes. Of course high-framerate cameras exist with similar, sometimes even superior, temporal performance, however they would require energy and memory budgets significantly higher than for event-cameras [8] and are not compatible with real-time operation at-the-edge.

A particularly exciting forward-looking goal is a multi-layer 3D-integrated smart imager chip whereby the event-camera is tightly integrated with an AI co-processor that can operate very effectively near the data-generating pixels. With the addition of an extra layer to a given 2-layer BSI imager, it would be possible to integrate AI capabilities, and in that case, specific AI acceleration adapted to event-based processing, to achieve in-sensor processing [9].

Historically there are two schools of thought for applying neural networks to event-based visual data: Convolutional Neural Networks (CNNs) and Spiking Neural Networks (SNNs). While CNNs convert events into static 2D frames, SNNs compute in an event-driven fashion similar to the sensor. While somewhat opposed, dedicated hardware implementations of both approaches harness model sparsity to compute more efficiently. In this article we briefly review recent advances in event-cameras in section II before addressing the convolutional versus spiking neural network dichotomy in detail in section III. In section IV we give some perspectives on an promising new alternative based on graph neural networks and conclude with a discussion in section V.

978-3-9819263-7-8/DATE23/© 2023 EDAA

## II. Trends in event-camera technologies

Event-camera technologies have rapidly undergone industrialization during the last decade. At the time of writing, there are four large players in the market: Prophesee [10], Samsung [11], Sony [10], and Omnivision [12] in addition to a host of smaller to medium sized start-ups and academic institutions [6], [13]–[15]. As a result, event-cameras have witnessed aggressive pixel pitch and array size scaling as observed in Fig.1. In particular, the incorporation of backside illuminated (BSI) processes and 3D wafer-stacking has permitted a considerable gain in the pixel fill factor - going from around one fifth to more than three quarters of the total area utilization - and pixel sizes starting to approach the range of conventional global-shutter pixels ($\leq 5\mu$m) [10], [11]. Steady improvements in throughput of the array readout systems, reaching the GEPS (gigaevents per second) range, allow to conserve the temporal precision of the pixel events at increasing array sizes [10]. The dual active and event pixel paradigm [13], [16] (i.e., allowing events and image data to be recorded simultaneously) has recently gained momentum again. While further miniaturization may become increasingly problematic, owing to the complexity of event-pixel circuits, alternatives based on emerging nanodevices could provide alternative solutions. For example, perovskite nanowires [17] and capacitors [18] as well as 2D hetrostructures [19] have been demonstrated, at array level, to generate events upon local illuminance changes - relying on device physics instead of active circuits.

High-resolution sensors can have side effects, as illustrated in [20]. Even though event sensors generate inherently sparse data, high rates can occur, in particular when the camera undergoes egomotion. Therefore the development of mitigation strategies such as in-sensor down-sampling [21], electronically foveated event-pixels [22] or centre surround [23] may be required. It remains to be seen what factors (i.e., further latency reduction, reduced power consumption, finer contrast sensitivity, greater dynamic range) may be the next key drivers in the development of the technology - these choices will most likely depend on which event-based computing paradigm begins to gain traction in real-world industrial scenarios.

## III. The CNN vs. SNN dichotomy

A promising and flexible solution for processing event-data is through the data-driven approach of neural networks - whereby the parameters that define how the model processes input are defined using a training procedure and a set of data. These parameters may either be learned *off-chip* (i.e., on a GPU server) and then transferred to the hardware system executing the model calculations near the event-camera. Otherwise they can be learned directly *on-chip* which promises to be essential in envisaged auto-adaptive systems capable of continually updating their operation to track data distribution changes and the emergence of new classes and objects of interest.

### A. Spiking neural networks

The most natural approach for processing event data would immediately appear to be that of SNNs. They have their roots

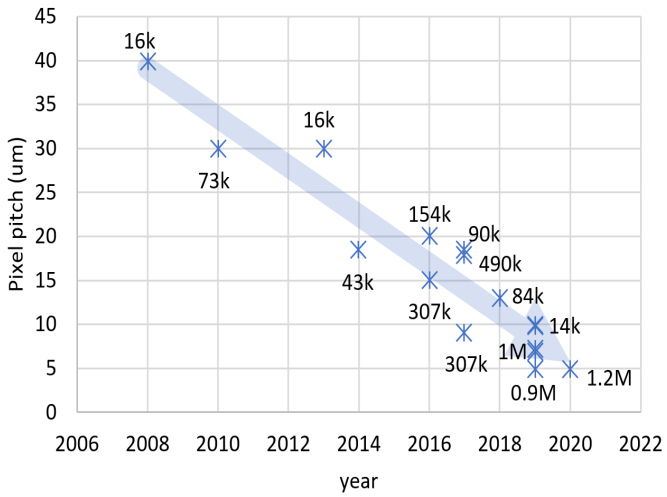


Fig. 1. Pixel size and array size trends over the decade for event-cameras.

in early research conducted in the mid-twentieth century based on the giant descending axon of the squid [24]. Neurons are modelled as integrating a weighted sum of their inputs into a dynamic state variable which often decays with a certain time constant. Neuron models can contain up to four differential equations, depending on the level of realism required by the designer. The Leaky-Integrate-and-Fire (LIF) neuron uses one equation to model the behaviour of the membrane potential of the neuron - corresponding to a simple resistor-capacitor circuit (Fig.2) and is the model of choice for most SNNs. Its simple mechanisms can derive mathematical equivalence with non-spiking neurons, are easy to implement in effective learning frameworks and offer lighter hardware implementations. SNN architectures most often take the form of multiple layers of LIF neurons whose neuron state variables are updated periodically with a certain timestep granularity (typically milliseconds).

Owing to their bio-inspired origins, the capability of SNNs to solve problems using hand-tuned coincidence detection architectures [25], [26] and to perform bio-inspired Hebbian learning [27] have been investigated. Although it has been extended to reinforcement learning [28], and in limited cases to supervised learning [29], modern SNNs are most often trained using the surrogate gradient method [30] (Fig.2). Here, the derivative of the spiking activation (a delta function that is zero everywhere besides at the spiking threshold) is replaced with a smooth function that approximates it. Loss functions based on the membrane potential [30], firing activity [31], time-to-first-spike [32] or temporal difference [33] of a population of neurons in the network output layer are often used. While these approaches may be satisfactory for off-chip learning scenarios, surrogate gradient backpropagation is an unrealistic algorithm for on-chip learning due to the prohibitive amount of memory that would be required to store the activity of all neurons over a potentially large number of timesteps. Approaches such as eligibility-propagation [34] and event-based random feedback alignment [31] are more realistic solutions whereby gradients can be approximated using neuron state variables without resorting to backpropagation. Other approaches also exist for off-chip

learning where SNNs are obtained through the conversion of a pre-trained neural network with continuous-valued outputs. Non-spiking neural networks are generally easier to train and scale better to more complex architectures such that event-cameras may be used not only for classification, but also for event-based segmentation and detection [35]. In order to achieve this, the activity of a spiking neuron is used as an approximation of a continuous value which can be achieved through a variety of encoding formats - most commonly rate-coding [36]. Although, this can result in excessively active neurons and unevenness error (when actual firing rate does not match the approximated value due to stimulation order). Conversion based on temporal-difference coding [37] or even by interpreting spikes as bits of digital words [38] can lead to sparser network activities. To facilitate this conversion, the non-spiking neurons are constrained to a low-precision integer number and trained using the straight-through estimator [39].

The sparse and event-driven nature of SNNs offers unique opportunities for innovative hardware design privileging low-power and low-latency operation. Furthermore, relative to neural networks with continuous valued neurons, SNNs avoid computing multiplications when evaluating weighted summations and instead relying on additions which require around four times less energy [40]. SNN accelerators, also referred to as neuromorphic processors, often group neurons in time-multiplexed cores. These SNN cores [41] are typically composed of separate neuron and synapse modules. Each contain a memory hierarchy (i.e., SRAM, standard cell memory and register files) which store information on the state of neurons and synapses and special purpose arithmetic logic units to calculate the evolving state variables of neurons and synapses. In such approaches memory accesses dominate energy consumption - as high as 99% of the total [42]. As a result, the fact that SNNs rely mainly on addition operations, instead of multiplication, is largely irrelevant. In the distributed core approach [43], each neuron and synapse in a neural network model are compiled onto a dedicated region of the chip - in the most extreme case with a one-to-one correspondence with physical circuits on the silicon. Principally this allows for the computing elements and the memory to be brought as close together as possible - ultimately reducing the cost of frequent memory access - although this typically degrades neuron density and results in a bigger silicon area and a higher cost for equivalent models. While digital hardware will typically update the weighted sums that are fed into the neurons in an event-driven fashion, the update procedure for neuron state variables and for generating neurons spikes is most often a clocked process that is triggered at regular intervals. While event-based state updates have been studied [44], they generally require more memory accesses, higher complexity calculations that ultimately leads to a less efficient implementation [42] and poor scalability. Digital neuromorphic processors arguably do not optimally exploit the event-based nature of the spiking neuron. Rather, analogue neuromorphic processors seem to be better adapted for seamless event-based operation [45]–[47]. Like early event-cameras, the objective is to harness the raw physical properties of transistors to mimic neuron and synaptic dynamical processes like leaky-integration, post-synaptic potentials, refractory behaviours and spike-frequency adaptation [48]. Crucially, unlike in the digital approach, time implicitly represents itself and state variables evolve naturally using the physics of the analogue circuit. A particularly interesting perspective is a fully-analogue system whereby the digital memories, used in current analogue processors, are replaced by emerging non-volatile memory technologies. This would permit multiplication and addition to be evaluated (using Ohm's and Kirchoffs laws) physically inside of the memory circuit itself [49] and for centralised bias generating units (which define neuron parameters) to be replaced by programmable conductance elements integrated directly into the circuits. However, as is the case with many analogue systems, transistor mismatch and other physical non-idealities limit the robustness of this approach.

### *B. Convolutional neural networks*

Like SNNs, activations (more commonly referred to as feature maps) in CNNs are also inherently sparse - in particular when used in combination with rectifying activation functions [50]. Furthermore, techniques such as pruning [51] and weight quantization [52] result in many zero-valued weights - making the CNN itself sparse. Unlike SNNs, however, CNNs are not immediately compatible with streaming event-data. 2D CNNs take as input stacked 2D matrices (e.g., the three red, green and blue channels in colour images) and therefore a pre-processing step is required convert the stream of events into a so-called *dense-frame*. The most simple solution is simply to count the number of generated events, per pixel, during a temporal window (typically tens to hundreds of milliseconds) [53], [54]. Negative polarity events can be subtracted from positive ones to create a single frame, or positive and negative events can accumulate in two separate channels (Fig.2). Some empirical results have even shown that CNNs, using event-data in this fashion, can achieve better performance than CNNs using standard frames [55]. However, this effectively discards the fine microsecond level temporal resolution of motion captured by the sensor. Other aggregation methods aim to preserve some of this information by making use of time surfaces [56] where pixel intensities encode the time since each pixel last generated an event. Some works also jointly use event counting and time surfaces together [57] or even train a recurrent neural network to generate frames based on the event-based input [58]. One disadvantage of these methods is that the possibility for event-driven computation is lost, since frames are prepared as periodic intervals. One solution to this may be through sub-manifold convolutions [59] whereby, as events arrive one at a time, only a subset of calculations are performed based on determining the active regions of affected feature maps in different layers.

One principal advantage of dense-frame CNN approaches is that they are immediately compatible with existing, highly optimized CNN accelerators. Such hardware typically fall into two categories : systolic processing element arrays and zero-skipping processors. Systolic processor arrays distribute computation (i.e., convolution of specific feature maps with specific kernels) over the array before spatially summing (between

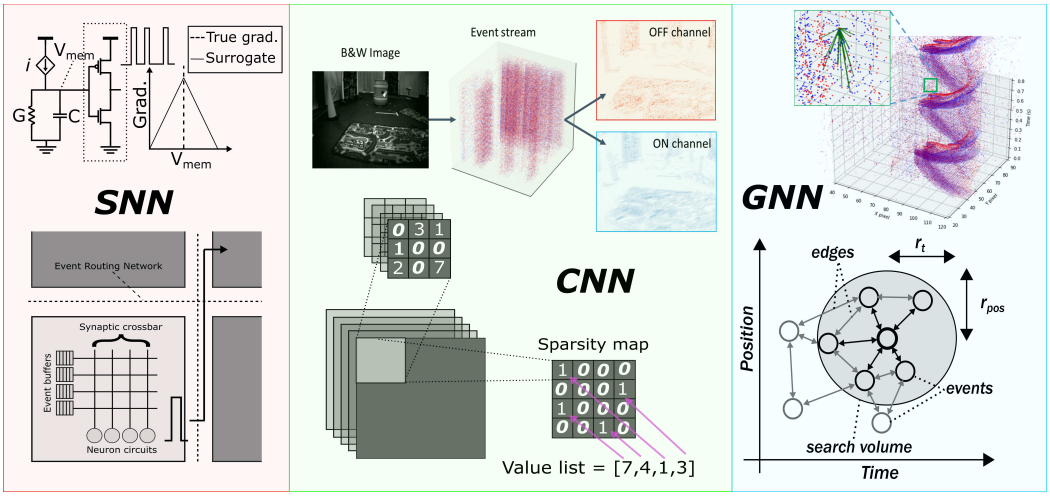


Fig. 2. Left (red) SNN: an example of the electrical circuit model of a spiking neuron and its surrogate gradient, an example of a neuromorphic spiking processor. Centre (green) CNN: an example of how a two-channel dense-frame is constructed from a series of events, sparse CNN feature maps and kernel weights and an example of how the feature map may be compressed. Right (blue) GNN: examples depicting how graphs are created from a set of events.

neighbouring elements) the resulting partial feature maps [60], [61]. While achieving massive parallelization and having a deterministic memory access pattern, they do not necessarily exploit CNN sparsity (i.e., the zeros within the convolutional feature maps and kernel weights) to reduce the amount of computation. Zero-skipping CNN accelerators, on the other hand, incorporate two main innovations to exploit CNN sparsity. As the name implies, the principal innovation is skipping multiplications by zero - ideally saving clock cycles. This can be achieved by skipping zero values in feature maps [62] or skipping zero-valued weights [63]. Some accelerators are capable of skipping both zeros in feature maps and weights at the expense of an increase in complexity [64]. The second principal innovation is the compressed format of the stored data which helps reduce memory accesses (Fig.2). However, this results in an inefficient non-deterministic SRAM access pattern. To mitigate this, CNNs may be trained with a set of constraints such that sparsity has a regular structure with reduced memory accesses [65]. It should be noted that structured sparsity is not only advantageous for zero-skipping but systolic processing element arrays too, and that both approaches exploit benefit from data reuse strategies where data is typically used several times for single memory access [66].

## IV. Are event-graphs the solution?

Recently, a third option for event-based AI using Graph Neural Networks (GNNs) [67], [68] has emerged as a contender. GNNs can learn data sharing and feature computation aspects in graphs. Considering a generated stream of events as a point-cloud in two spatial and one temporal dimensions, a graph can be constructed by, for example, connecting events through directed edges based on their euclidean distance. Layers of graph convolutions can then be applied in order to find increasingly powerful representations for each event. Since graph edges allow for spatiotemporal differences between events to be incorporated into the convolutions, graph convolutions can exploit the precise timing information captured by an event-camera deep into a neural network. Like SNNs and recurrent CNNs, they also naturally integrate information from the past (and future) into their current state as new events are continually incorporated. Event-GNNs have already outperformed dense-frame CNNs on a variety of event-camera benchmarks in classification [69], object detection [70], segmentation [71] and optical-flow estimation [72] while remarkably requiring orders of magnitude fewer neural network calculations and parameters. Event-graphs are also inherently sparse and amenable to event-driven operation because graph convolutions could be triggered upon the generation of each event. Despite this early promise, there remain numerous roadblocks that need to be removed before event-graphs can realise their potential - in particular there is a hardware vacuum. While dedicated GNN accelerators have recently been proposed [73], [74] for datacenters, they are poorly adapted for the sparse streaming nature of event-data and low-power operation at the edge. Perhaps most problematic of all is the latency required to incorporate events into a continuously evolving event-graph (generally based on tree-search methods [75]) - although algorithmic innovations have already resulted in a four order of magnitude speed-up [72] that brings closer the possibility of real-time event-graph processing.

## V. Discussion

The motivation for SNNs in the papers included in this review is, that they are sparse and event-driven and therefore will ultimately be well suited for low-power edge AI systems. Current SNN hardware, however, is largely clock-based, and CNNs, due to pruning, rectifying activation functions and weight quantization, are also highly sparse. In some cases, the inverse is in fact true and digital CNN hardware implementations are more efficient than digital SNNs [42]. While it may be argued that SNNs are required for tasks relying on temporal memory, recurrent blocks can be readily incorporated into CNNs for this purpose, too [76]. Furthermore, SNNs have been observed to consistently exhibit a degraded performance relative to CNNs when applied to a variety of event-camera benchmarks [77]. This conclusion may feel somewhat unsatis-

| Near EB Sensor | SNN | CNN | GNN |
|---|---|---|---|
| Data - Exploit temporal information | ++ | - | ++ |
| Data - Sparsity | ++ | - | ++ |
| Data - Preparation (↓) | ++ | + | - - |
| Computation - Sparsity | ++ | + | ++ |
| Computation - # Operations (↓) | + | - | ++ |
| Application - Accuracy | - | + | ++ |
| Hardware - Maturity | + | ++ | - - |
| Memory - Footprint (↓) | + | ++ | ? |
| Memory - Bandwidth (↓) | + | - | ? |
| System - Energy Efficiency | ++ | + | ? |
| System - Configurability / Scalability | - | ++ | ++ (?) |
| System - Latency (↓) | ++ | - | ++ (?) |

+ stands for "has better metrics in". ↓ lower is better

TABLE I
QUALITATIVE COMPARISON TABLE.

factory: How can the best way of treating event-data be through discarding the temporal information?

In practical evaluations, CNN accelerators [62] and digital spiking neuromorphic processors [78] exhibit power consumption of the order of hundreds of milliwatts (although these vary with network size and sparsity), while analogue spiking processors generally consume an order of magnitude less power [46]. These systems may therefore be advantageous in applications where energy is extremely scarce and high task accuracy is of secondary importance.

On the other hand, SNNs have the advantage of being fully event-driven enabling low-latency systems and are immediately compatible with the address-event representation protocols that are already in use at the sensor. CNNs largely lack this potential for data-driven computation that puts a lower bound on, for example, how fast they can respond to changes in their input data. Thus, SNN appear to be the natural choice for exploiting the time-domain information, and consequently high temporal resolution, of event-cameras, particularly in vision tasks requiring optimized system response latency. SNNs may also have a greater potential with regards to efficient on-chip learning by exploiting event-triggered and backpropagation-free gradient approximation techniques which are supported in recent neuromorphic processors [41]. They may be best suited for scenarios therefore where the system will be required to continually learn and update its operation over time without the possibility of off-chip retraining.

A solution to forego the above summarized conflicts may reside in the exciting new research into event-graph neural networks which, like SNNs, compute in an event-driven fashion. Rather than discarding spatiotemporal information, event-graphs incorporate it into their edges and use it to perform graph convolutions and ultimately appear capable of outperforming CNNs with substantial reductions in memory and calculation resources.

New neuromorphic event-graph hardware, which does not exist today, will need to be developed in order for this elegant data-driven approach to fulfill its potential and we expect this to emerge as a new active area of research in coming years.

## REFERENCES


[1] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems*. Springer Science & Business Media, 1989, vol. 80.

[2] T. Delbrück and C. Mead, "An electronic photoreceptor sensitive to small changes in intensity," *Advances in neural information processing systems*, vol. 1, 1988.

[3] M. A. Mahowald, "Silicon retina with adaptive photoreceptors," in *Visual information processing: from neurons to chips*, vol. 1473. SPIE, 1991, pp. 52–58.

[4] A. Dickinson *et al.*, "A 256/spl times/256 cmos active pixel image sensor with motion detection," in *ISSCC*. IEEE, 1995, pp. 226–227.

[5] E. Culurciello and R. Etienne-Cummings, "Second generation of high dynamic range, arbitrated digital imager," in *2004 IEEE International Symposium on Circuits and Systems*, vol. 4. IEEE, 2004, pp. IV–828.

[6] P. Lichtsteiner *et al.*, "A 128×128 120 db 15$\mu$ s latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[7] C. Zamarreño-Ramos *et al.*, "Multicasting mesh aer: A scalable assembly approach for reconfigurable neuromorphic structured aer systems. application to convnets," *IEEE transactions on biomedical circuits and systems*, vol. 7, no. 1, pp. 82–102, 2012.

[8] T. G. Etoh *et al.*, "Toward one giga frames per second—evolution of in situ storage image sensors," *Sensors*, vol. 13, no. 4, pp. 4640–4658, 2013.

[9] P. Vivet *et al.*, "Advanced 3d technologies and architectures for 3d smart image sensors," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 674–679.

[10] T. Finateu *et al.*, "A 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *ISSCC*, 2020, pp. 112–114.

[11] Y. Suh *et al.*, "A 1280× 960 dynamic vision sensor with a 4.95-$\mu$m pixel pitch and motion artifact minimization," in *2020 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[12] S. Chen and M. Guo, "Live demonstration: Celex-v: A 1m pixel multi-mode event-based sensor," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2019, pp. 1682–1683.

[13] C. Brandli *et al.*, "A 240× 180 130 db 3 $\mu$s latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[14] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128× 128 1.5% contrast sensitivity 0.9% fpn 3 µs latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *JSSC*, vol. 48, no. 3, pp. 827–838, 2013.

[15] M. Akrarai *et al.*, "An asynchronous hybrid pixel image sensor," in *2021 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2021, pp. 55–61.

[16] C. Posch *et al.*, "A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, 2010.

[17] L. Gu *et al.*, "A biomimetic eye with a hemispherical perovskite nanowire array retina," *Nature*, vol. 581, no. 7808, pp. 278–282, 2020.

[18] C. Trujillo Herrera and J. G. Labram, "A perovskite retinomorphic sensor," *Applied Physics Letters*, vol. 117, no. 23, p. 233501, 2020.

[19] Z. Zhang *et al.*, "All-in-one two-dimensional retinomorphic hardware device for motion detection and recognition," *Nature Nanotechnology*, vol. 17, no. 1, pp. 27–32, 2022.

[20] D. Gehrig and D. Scaramuzza, "Are high-resolution event cameras really needed?" *arXiv preprint arXiv:2203.14672*, 2022.

[21] M. Bouvier *et al.*, "Scalable pitch-constrained neural processing unit for 3d integration with event-based imagers," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 385–390.

[22] T. Serrano-Gotarredona and B. Linares-Barranco, "System architectures for electronically foveated dynamic vision sensor," in *2022 37th Conference on Design of Circuits and Integrated Circuits (DCIS)*. IEEE, 2022, pp. 01–06.

[23] T. Delbruck *et al.*, "Utility and feasibility of a center surround event camera," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 381–385.

[24] J. Z. Young, "The functioning of the giant nerve fibres of the squid," *Journal of Experimental Biology*, vol. 15, no. 2, pp. 170–185, 1938.

[25] T. Dalgaty *et al.*, "Insect-inspired elementary motion detection embracing resistive memory and spiking neural networks," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2018, pp. 115–128.

[26] F. Moro *et al.*, "Neuromorphic object localization using resistive memories and ultrasonic transducers," *Nature communications*, vol. 13, no. 1, pp. 1–13, 2022.

[27] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.

[28] W. Gerstner *et al.*, "Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules," *Frontiers in neural circuits*, vol. 12, p. 53, 2018.

[29] Y. Hao *et al.*, "A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule," *Neural Networks*, vol. 121, pp. 387–395, 2020.

[30] E. O. Neftci *et al.*, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[31] ——, "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Frontiers in neuroscience*, vol. 11, p. 324, 2017.

[32] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 3227–3235, 2017.

[33] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[34] G. Bellec *et al.*, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature communications*, vol. 11, no. 1, pp. 1–15, 2020.

[35] S. Kim *et al.*, "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 270–11 277.

[36] P. U. Diehl *et al.*, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International joint conference on neural networks (IJCNN)*. ieee, 2015, pp. 1–8.

[37] B. Rueckauer and S.-C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," in *2018 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[38] ——, "Temporal pattern coding in deep spiking neural networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[39] Y. Bengio *et al.*, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[40] A. Pedram *et al.*, "Dark memory and accelerator-rich system optimization in the dark silicon era," *IEEE Design & Test*, vol. 34, no. 2, pp. 39–50, 2016.

[41] C. Frenkel and G. Indiveri, "Reckon: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales," in *ISSCC*, vol. 65. IEEE, 2022, pp. 1–3.

[42] M. Dampfhoffer *et al.*, "Are snns really more energy-efficient than anns? an in-depth hardware-aware study," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2022, pp. 1–11, 2022.

[43] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[44] J. Stuijt *et al.*, "$\mu$brain: An event-driven and fully synthesizable architecture for spiking neural networks," *Frontiers in neuroscience*, vol. 15, p. 538, 2021.

[45] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[46] S. Moradi *et al.*, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps)," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 106–122, 2017.

[47] C. Pehle *et al.*, "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, 2022.

[48] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.

[49] T. Dalgaty *et al.*, "In situ learning using intrinsic memristor variability via markov chain monte carlo sampling," *Nature Electronics*, vol. 4, no. 2, pp. 151–161, 2021.

[50] M. Kurtz *et al.*, "Inducing and exploiting activation sparsity for fast inference on deep neural networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5533–5543.

[51] P. Molchanov *et al.*, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[52] A. Zhou *et al.*, "Incremental network quantization: Towards lossless cnns with low-precision weights," *arXiv preprint arXiv:1702.03044*, 2017.

[53] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," 2018.

[54] D. Gehrig *et al.*, "End-to-end learning of representations for asynchronous event-based data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5633–5643.

[55] A. I. Maqueda *et al.*, "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*, 2018, pp. 5419–5427.

[56] A. Sironi *et al.*, "Hats: Histograms of averaged time surfaces for robust event-based object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740.

[57] A. Z. Zhu *et al.*, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[58] M. Cannici *et al.*, "A differentiable recurrent surface for asynchronous event-based data," in *European Conference on Computer Vision*. Springer, 2020, pp. 136–152.

[59] N. Messikommer *et al.*, "Event-based asynchronous sparse convolutional networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 415–431.

[60] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[61] M. Lepecq *et al.*, "End-to-end implementation of a convolutional neural network on a 3d-integrated image sensor with macropixel array," *In Submission*, 2023.

[62] A. Aimar *et al.*, "Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 644–656, 2018.

[63] S. Zhang *et al.*, "Cambricon-x: An accelerator for sparse neural networks," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2016, pp. 1–12.

[64] Y.-H. Chen *et al.*, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 292–308, 2019.

[65] Z.-G. Liu *et al.*, "S2ta: Exploiting structured sparsity for energy-efficient mobile cnn acceleration," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2022, pp. 573–586.

[66] Y.-H. Chen *et al.*, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[67] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[68] M. Fey *et al.*, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 869–877.

[69] Y. Bi *et al.*, "Graph-based object classification for neuromorphic vision sensing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 491–501.

[70] S. Schaefer *et al.*, "Aegnn: Asynchronous event-based graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 371–12 381.

[71] A. Mitrokhin *et al.*, "Learning visual motion segmentation using event surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 414–14 423.

[72] T. Dalgaty *et al.*, "Hugnet: Hemi-spherical update graph neural network for low-latency event processing," *Under submission*, 2023.

[73] S. Liang *et al.*, "Engn: A high-throughput and energy-efficient accelerator for large graph neural networks," *IEEE Transactions on Computers*, vol. 70, no. 9, pp. 1511–1525, 2020.

[74] M. Yan *et al.*, "Hygcn: A gcn accelerator with hybrid architecture," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 15–29.

[75] K. Zhou *et al.*, "Real-time kd-tree construction on graphics hardware," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, pp. 1–11, 2008.

[76] E. Perot *et al.*, "Learning to detect objects with a 1 megapixel event camera," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 639–16 652, 2020.

[77] R. Baldwin *et al.*, "Time-ordered recent event (tore) volumes for event cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[78] P. Blouw *et al.*, "Benchmarking keyword spotting efficiency on neuromorphic hardware," in *Proceedings of the 7th annual neuro-inspired computational elements workshop*, 2019, pp. 1–8.