Intelligent Computing
A SCIENCE PARTNER JOURNAL

REVIEW ARTICLE

Unconventional Integrated Photonic Accelerators for High-Throughput Convolutional Neural Networks

Aris Tsirigotis[1], George Sarantoglou[1], Menelaos Skontranis[1], Stavros Deligiannidis[2], Kostas Sozos[2], Giannis Tsilikas[3], Dimitris Dermanis[1], Adonis Bogris[2*], and Charis Mesaritakis[1]

[1]Department of Information and Communication Systems Engineering, Engineering School, University of the Aegean, Samos, Greece. [2]Department of Informatics and Computer Engineering, University of West Attica, Egaleo, Greece. [3]School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Athens, Greece.

*Address correspondence to: abogris@uniwa.gr

Citation: Tsirigotis A, Sarantoglou G, Skontranis M, Deligiannidis S, Sozos K, Tsilikas G, Dermanis D, Bogris A, Mesaritakis C. Unconventional Integrated Photonic Accelerators for High-Throughput Convolutional Neural Networks. *Intell. Comput.* 2023;2:Article 0032. https://doi.org/10.34133/icomputing.0032

Submitted 2 November 2022
Accepted 9 May 2023
Published 23 June 2023

We provide an overview of the rapidly evolving landscape of integrated photonic neuromorphic architectures, specifically targeting the implementation of convolutional neural networks. The exploding research momentum stems from the well-known advantages of photonic circuits compared to digital electronics, and at the same time, it is driven by the massive need for cognitive image/video processing. In this context, we provide a detailed literature review on photonic cores operating as convolutional neural networks, covering either the functionality of a conventional neural network or its spiking counterpart. Moreover, we propose 2 alternative photonic approaches that refrain from simply transferring neural network concepts directly into the optical domain; instead, they focus on fusing photonic, digital electronic, and event-based bioinspired processing to optimally exploit the virtues of each scheme. These approaches can offer beyond state-of-the-art performance while relying on realistic, scalable technology. The first approach is based on a photonic integrated platform and a bioinspired spectrum-slicing technique. The photonic chip allows feature extraction through optical filtering with low power consumption and an equivalent computational efficiency of 72 femtojoules per multiply-and-accumulate operation for 5-bit precision. When combined with typical digital neural networks, an almost 5-fold reduction in the number of parameters was achieved with a minor loss of accuracy compared to established convolutional neural networks. The second approach follows a bioisomorphic route in which miniaturized spiking laser neurons and unsupervised bioinspired training are unified in a deep architecture, revealing a noise-resilient and power-efficient proposition.

Introduction

Brain-inspired computational paradigms have paved the way for applications relying on perception and adaptability. The most successful brain-inspired paradigm is the artificial neural network (ANN), which mimics the multilayer architecture of biological neural networks at both the operational and structural levels [1]. Training algorithms should be used to unleash their true potential. One of the most successful examples that unfortunately seem to lack biocompatibility is the backpropagation algorithm [2], which can tune an ANN to extract meaningful features from raw data in applications such as pattern recognition and decision-making, a procedure known as representation learning [1]. These raw data streams are highly heterogeneous and are generated by an ever-expanding palette of applications such as autonomous vehicles, robotics, medical diagnostic tools, and multisensory systems. Despite their diversity, these data streams often involve images or videos that require cognitive image processing. The most successful type of ANN for image processing is a convolutional neural network (CNN) [3]. Through weight sharing and convolution attributes, CNNs can process images while minimizing the computational overhead of typical ANNs, which is mainly attributed to fully connected networks. Initially, CNNs were limited to simple tasks, such as recognizing handwritten digits [3], owing to the lack of powerful computational resources and sufficiently large training datasets [1]. The landscape changed with the demonstration of a deep multilayer CNN known as AlexNet, implemented on a graphics processing unit (GPU) to handle the ImageNet dataset [4]. This work accomplished the resurgence of the field, leading to the development of more complex deep CNNs such as ZFNet [5], VGGNet [6], GoogleNet [7], and ResNet [8]. Advances in deep CNNs have provided state-of-the-art results in applications such as image recognition [4,8], object detection [9,10], and speech recognition [11].

Although an increase in depth and trainable parameters for CNNs results in better performance, it also imposes an equally important increase in power consumption and memory demand. A survey by OpenAI shows that from 2012 (AlexNet) to 2018

(AlphaZero [12]), an increase in the number of computations by a factor larger than 300,000 was observed. In contrast, in the same period, Moore's law accounts for only a 7-fold increase in computational power [13]. More specifically, in image processing, the convolution stages of CNNs account for 80% of the total power consumption [14]. Consequently, to meet the exponentially increasing demand, companies and researchers aim to exploit multiple chips and unlock massively parallel computations. This leads to a severe increase in the energy footprint, thereby raising important financial and ecological concerns when upscaling is considered [15].

The fundamental difference between ANN implementation and the actual operation of biological neural circuits is a potential source of scalability bottlenecks in ANN implementation. These differences can be traced both at the "algorithmic level" and, more importantly, in the underlying physical hardware responsible for the "computations." In light of this, the first route toward mitigation of these effects targets the algorithmic architecture of ANNs and corresponds to closing the gap between processing in the brain and ANNs as much as possible. In particular, ANNs are based on the crude oversimplification of biological architectures, which dictates the substitution of spiking neurons with non-linear functions under the rate-encoding hypothesis [16]. Spiking is a sparse spatiotemporal encoding scheme present in the brain that partly accounts for the large power efficiency of the brain [17]. Information encoding is still a matter of scientific debate; however, the main hypothesis dictates that information is encoded at the firing/arrival time of pulses with relatively constant amplitude and temporal width, which are generated by ionic interactions on the membrane of neurons [16]. In neuroscience, these pulses are known as activation potentials or spikes. In artificial spiking neural networks (SNNs), neurons are modeled by non-linear systems such as the leaky integrate and fire (LIF) [18], Izhiekevich [18], and spike response models (SRM) [19], which exhibit the property of excitability and entail major neurocomputational attributes such as thresholding, a refractory period, potentiation, and inhibition [18]. Multiple spiking codes have been proposed to explain how information is represented inside the brain, which can be harnessed by SNNs to significantly decrease power consumption [17]. This spike-driven paradigm shift unlocks event-based processing because computations are performed only when a spike is available at the input of a neuron. SNNs implemented on regular platforms require fewer hardware resources and achieve faster processing than ANNs [20,21]. The major reason behind the dominance of ANNs in the technological landscape is that SNNs are challenging in terms of training because they are incompatible with the aforementioned backpropagation algorithm. However, this picture can change because there is strong ongoing research focusing either on the exploitation of biologically plausible algorithms based on Hebbian learning strategies, such as spike timing-dependent plasticity (STDP) [22], or on the adaptation of the backpropagation algorithm in the case of SNNs [23]. SNNs can implement CNNs (SCNN) by simply following the interconnection rules dictated by CNNs [24,25].

Another route toward energy-efficient computation lies in the development of novel platforms that can mimic the structural architecture of the brain at the hardware level. Both ANNs and SNNs are typically implemented on conventional von Neumann computers, where the memory and processing units are physically separated, thus demanding costly data transfer, an issue known as the von Neumann bottleneck [26]. The von Neumann processors are significantly burdened when ANNs are implemented. This stems from the fact that the multiply-and-accumulate (MAC) operations required to compute the flow of information through the synaptic connections in a single fully connected layer (FCL) with N neuron scale with $O(N^2)$ compared to $O(N)$ for non-linear activation. This is in contrast with the processing methodology of biological neurons, where there is no such separation because memory is implemented by synaptic weights, and processing is performed by the non-linear transformation of the neural cells. These 2 processes are collocated, thereby alleviating the inherent non-von Neumann constraints. Therefore, dedicated hardware that directly mimics the architecture of neural networks has been developed. A characteristic case is the realization in recent years of multiple digital neuromorphic platforms, such as True North [27], SpiNNaker [28], BrainScaleS [29], and Loihi [30], which aim to combine brain-inspired architectural rules with spiking coding schemes. Another important direction is the design of analog electronic hardware to directly mimic neural structures, such as crossbars based on resistive elements [31]. Although analog processors suffer from low bit precision compared to digital processors, research on neural network performance has revealed that high bit precision is not always imperative [32,33]. However, despite the disruptive nature of these attempts, their full-scale applicability and technological maturity remain limited. Therefore, these analog bioinspired devices are mainly designed and utilized as computational accelerators, meaning that they aim to unburden conventional processors from the demanding computational bandwidth required at the first neural layers and not to tackle the entire processing pipeline, as illustrated in Fig. 1.

Photonics is a rapidly evolving field that harnesses the unique properties of light for transmitting and processing information and has the potential to revolutionize various applications, including data communication, sensing, and computing. Unlike electrons, photons do not experience resistance or capacitance, and can travel without signal degradation. This feature enables photonic systems to operate at significantly higher speeds and over longer distances than traditional electronic systems. Furthermore, photonics offers high bandwidths, which enable the rapid transfer of large amounts of data. Photonic devices can simultaneously transmit multiple signals, allowing high-bandwidth data transmission [34]. Additionally, photonic integration offers high levels of scalability and miniaturization, allowing the creation of compact, lightweight, and portable devices that can be easily integrated into existing systems. The potential of photonic integrated chips (PICs) has been demonstrated in various fields, such as optical communication, data centers, sensing, and biomedical systems.

However, several challenges and limitations must be addressed before integrated photonics can become a viable alternative to their electronic counterparts. One of the main challenges is integrating different photonic components into a single device. Unlike electronic devices, which can be easily fabricated on a single substrate, photonic components often require different materials and fabrication processes. For instance, waveguides and detectors are typically fabricated using different materials and processes, making it challenging to integrate them into a single device. Moreover, photonic devices require materials that can efficiently interact with light. This restricts the range of materials that can be used for photonic integration, making it difficult to

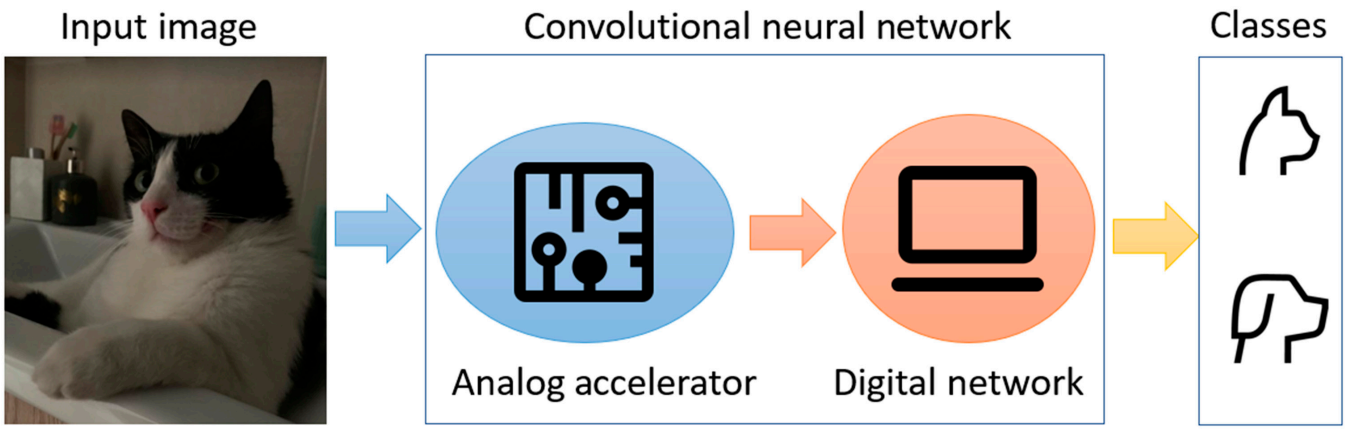

Fig. 1. Hybrid CNN structure for image classification. The first neural layers are implemented with an analog accelerator (electronic, photonic, etc.) with low power consumption and high processing bandwidth, whereas the last layers are implemented digitally with high precision and low complexity.

achieve the desired functionality. Another well-known limitation is the low integration scale of photonics owing to the need for precise alignment and control of light at the nanoscale level. This limitation can reduce the complexity and functionality of photonic devices. In addition, there is a lack of standardization in the design and fabrication of photonic components, which can hinder their adoption in commercial applications [35]. Despite these challenges and decade-long advancements in photonic component design, fabrication processes, and materials, integrated photonics technology has the potential to move beyond the well-established data-transfer role and infiltrate new industries such as high-performance computing and machine learning.

In the machine learning landscape, PICs have emerged as highly promising platforms for analog data processing [36]. Photonics offers 2 major advantages for addressing the core issues of neural-like computations: straightforward computational parallelism and linear operation at the speed of light. Parallelism in photonics can be achieved through wavelength-division multiplexing (WDM) or time-division multiplexing (TDM). In this case, the same physical device can process data encoded at different wavelengths/time slots, thus significantly increasing the computational density. Furthermore, compared to conventional electrical synapses (connections between neurons), photonics offers lower propagation losses and is free from electrical resistance-related thermal effects, which can be deleterious at high synaptic densities. More importantly, photonics does not suffer from the well-studied bandwidth/fan-in trade-off that plagues electronics [36]. For these reasons, photonic neural networks are attractive for high-bandwidth applications such as wideband radio frequency signal processing [37], fiber optic communications [38,39], non-linear programming in autonomous vehicles/robotics [40], and ultrafast online learning [41]. In terms of linear operations, by using specialized optical hardware structures, the equivalent of the MAC operations defined in electronics can be performed by a simple propagation of information at the speed of light through a passive optical mesh. Thus, a substantial increase in energy efficiency can reach up to a few femtojoules per multiply-and-accumulate operation (fJ/MAC) (3 orders of magnitude lower than the current GPUs) [34,42,43]. Finally, numerous excitable photonic devices can be used as spiking nodes in photonic SNNs [44]. Compared to biological and electronic neurons that produce spikes on the millisecond scale, photonic spiking neurons can produce optical spikes with durations in the nano- to picosecond-scale regime, with marginal power consumption owing to their high wall-plug efficiency, which renders them suitable for ultrafast event-based processing.

This study's aim is twofold: on the one hand, to provide an overview of current integrated photonic CNNs that tackle the demanding field of ultrafast image processing and, on the other hand, to present 2 alternative CNN photonic accelerators that offer a different perspective in this rapidly expanding field. In the Convolutional Neural Networks—Background section, a short introduction to CNNs is provided. The Convolutional Neural Networks on Photonic Integrated Platforms section provides a detailed review of the CNNs implemented in integrated photonic platforms. The Unconventional Convolutional Processing Based on Optical Spectrum Slicing section introduces an unconventional neuromorphic processor that leverages optical spectrum slicing (OSS) to perform convolutional processing in the analog domain. In the Photonic Spiking Convolutional Neural Networks section, recent advancements in spiking CNN accelerators are presented. The Generic Spiking Convolutional Neural Network for Image Classification section and the Deep Photonic Spiking Convolutional Neural Networks (Deep Spiking CNN) section introduce a different approach to photonic CNNs that targets the edge-to-edge training of a deep photonic spiking CNN using unsupervised local learning. The paper concludes with a discussion of the merging of OSS with a spiking operation for ultrafast and hardware-friendly convolutional and event-based processing.

Convolutional Neural Networks—Background

Inspired by visual information processing in the primary cortex of mammalian brains [45], a CNN is a type of ANN that utilizes convolutional operations instead of general matrix multiplications in at least one of its layers to extract diverse features from an input tensor (e.g., an image or video). The convolutional operation involves sliding a small filter, also known as a kernel, over the input tensor and computing the dot product between the filter and the pixel block, namely, the patch, on which it is currently positioned. Therefore, CNNs adopt the multilayer hierarchy of typical ANNs but introduce a succession of interlaying convolutional and pooling layers, followed by typical FCLs. While the convolutional layers aim to extract relevant features

from the input images, the pooling layers aim to compress the outputs of the convolutional layers, namely, the feature maps. This procedure reduces redundant spatial information and renders the network more robust to small deviations. Finally, the FCLs combine these features to form more complex patterns that allow for data classification [1]. A characteristic example of such an architecture is depicted in Fig. 2, which presents the LeNet-5 network [3]. Its convolutional layer consists of multiple kernels, each of which allows the extraction of different spatial features from the incoming images. The convolution process is illustrated in detail in Fig. 3. Each kernel is a small matrix that scans the image with a pixel step equal to S, known as the stride. At each position, the kernel values are multiplied element-wise by a patch, which is a matrix containing the pixel values of the underlying region. The elements of the resulting matrix were accumulated to produce a single value. In the context of neural networks, this value describes the correlation between the patch and the kernel [16]. The result is that each kernel filters the whole image with respect to a different visual pattern (feature). In detail, the application of K kernels with dimensions $N_k \times N_k$ and a stride S on a $W \times W$ image returns K lower-dimensional images with dimensions $\left(\frac{W-N_k}{S}+1\right) \times \left(\frac{W-N_k}{S}+1\right)$ The resulting elements pass through a non-linear function, such as a rectified linear unit (ReLU), and the resulting outputs are called feature maps. This non-linear activation introduced non-linearity into the output, allowing the network to learn complex and abstract features from an input tensor. Thus, each feature map contains information regarding the different spatial features of the original image. As previously mentioned, the convolutional procedure was inspired by image processing in the primary virtual cortex (V1 region), which is the first section of the visual pathway. In the V1 region, groups of neurons have receptive fields that perform visual filtering over different orientations [45], a process analogous to convolution between a kernel and an image [46–48].

Next, a pooling layer is applied to each feature map (the product of the kernel and patch) to perform dimensionality reduction in the form of compression. A window similar to the kernel is defined; however, in contrast to the convolution procedure, either the maximum or the average of the underlying patch is computed. The first method is known as max pooling, and the second method is known as average pooling. Dimensionality reduction is important because it renders the network insensitive to irrelevant variations and attributes with respect to inspected classes. The resulting output maps can either be driven to the next convolution-pooling stage to be filtered by a different set of kernels or to a typical FCL. The FCL(s) is typically located at the end of the CNN and is responsible for classifying the input image into several possible categories.

Spiking variations of CNNs follow the same architecture as their conventional counterparts; however, they use non-linear excitable systems such as LIF or SRM models. The major challenge with these networks is training because they do not directly support the backpropagation algorithm [23]. Spiking trains are described as a set of Dirac functions that are undifferentiable and, consequently, incompatible with the computation of gradients in the backpropagation algorithm. Furthermore, backpropagation is considered biologically implausible in neuroscience, mainly because of the lack of symmetric backward flow in biological networks, which is present in all neural connections [23]. A class of spiking CNNs performs unsupervised

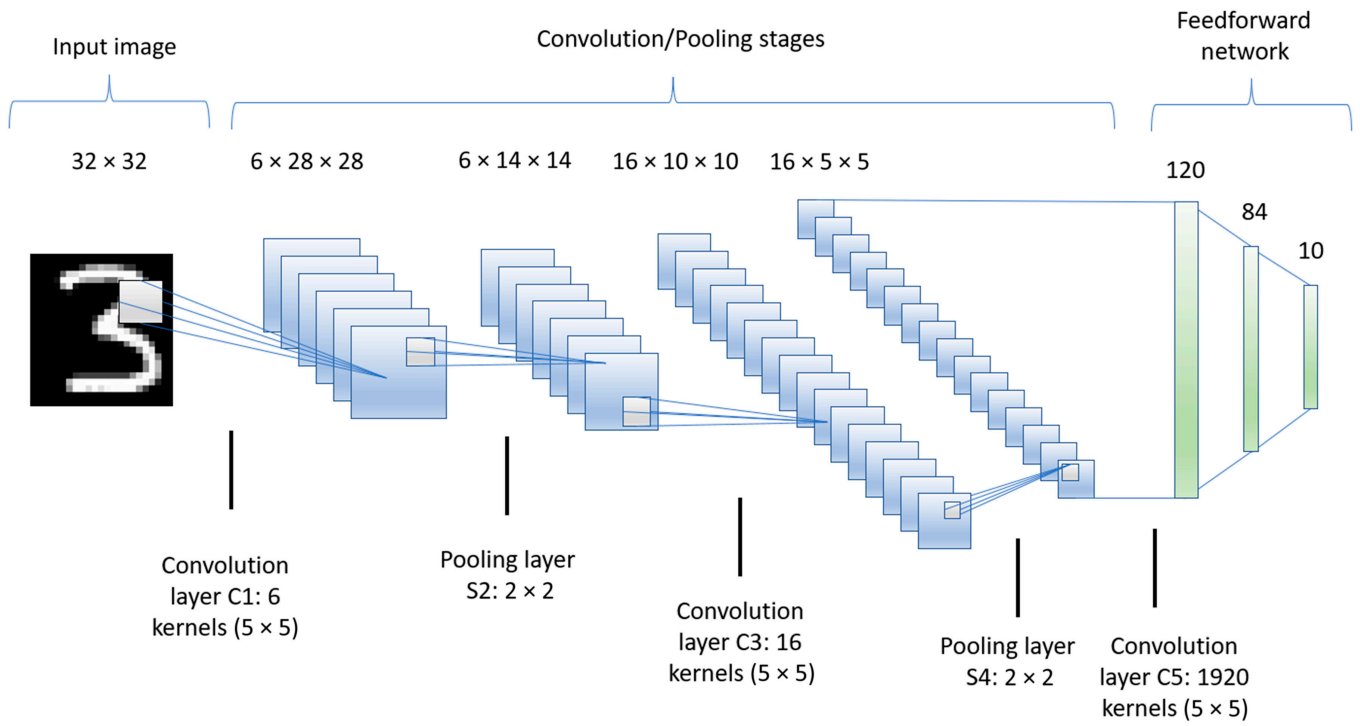

Fig. 2 . LeNet-5 CNN architecture. The network comprises convolutional, subsampling (pooling), and fully connected layers. The convolutional layers extract features from the input image using kernels, while the pooling layers reduce the spatial size of the feature maps. The fully connected layers classify the features into the desired output. Non-linear activation functions are used to introduce non-linearity in the output of the convolutional layers.

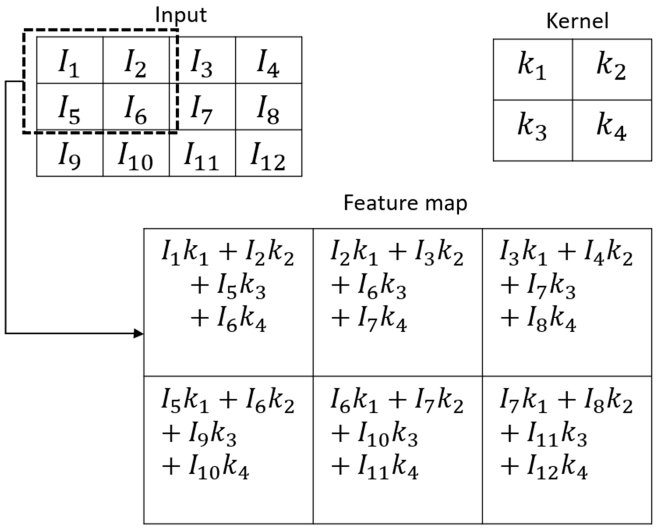

Fig. 3. Illustration of the convolutional process in a CNN with a 2 × 2 kernel and stride of 1. The kernel is applied to the input image in a sliding window manner, performing a dot product between the kernel and the input image pixels at each position. The resulting values are summed and produce a single output pixel in the feature map.

local learning using a biologically plausible STDP rule [24,25,49]. This rule is a part of the Hebbian learning algorithms, which state that "neurons that fire together, wire together". In particular, if a pre-synaptic spike arrives before the generation of a post-synaptic spike at the synapse, a causal connection is deduced and the synaptic weight is increased, an effect known as long-term potentiation. However, if the pre-synaptic spike arrives after the post-synaptic spike, then a low correlation between the 2 neurons is deduced, and the synaptic weight value is decreased, an effect known as long-term depression. However, the major drawback of STDP is that it constitutes an unsupervised learning algorithm and, as a result, is incompatible with supervised learning methods.

A different class of spiking CNNs focuses on the direct application of backpropagation using neural models, such as SRM, which provide the membrane potential at each node as the signal to be differentiated [50,51]. The best results in terms of accuracy are provided by spiking CNNs, which result from the conversion of pretrained conventional CNNs [21,52,53]. Although these converted networks are easier to train, they are restricted to a single encoding scheme—rate encoding—in which information is encoded at the firing rate of the neurons. This scheme is not as efficient as other encoding strategies in terms of sparsity, because information is carried by multiple spikes instead of one [20]. Additionally, this is not biologically plausible, because the firing rate cannot explain important processes in the visual pathways of mammals [17,54]. However, compared with conventional CNNs, they are still more power-efficient and have lower computational latency.

Convolutional Neural Networks on Photonic Integrated Platforms

Integrated photonic CNNs employ a procedure that transforms the convolution stage into a sequence of matrix vector multiplications. This method is inspired by the GPU implementations of CNNs [55] and is relevant in the case of resistive crossbar arrays [31]. In this algorithm, an image or feature map is projected onto a new matrix designed to contain a serialized patch in each row. The kernels were congregated in another matrix, with each column corresponding to the weight of each kernel. By multiplying the 2 matrices, the convolution between the input image and kernels is obtained. In photonics, this procedure is performed in a similar manner, where the patch/vector is imprinted on an optical carrier through electro-optic modulation; thus, the image vector is mapped to an optical time series that, in turn, is injected into a PIC to achieve matrix multiplication. In the case of reprogrammable photonics, the PIC is fabricated or programmed [56,57] to implement a matrix containing kernel weights. For each time step, a point for each feature map was obtained, which corresponded to the interaction between the patch and kernels. In this context, any PIC configured to perform multiplications can be considered a photonic CNN engine [42,58–60]. In the literature, 2 major categories of PICs implement such schemes: incoherent and coherent meshes [61].

With respect to optically incoherent meshes, a PIC has recently been experimentally demonstrated to perform convolutions at a high throughput by exploiting wavelength multiplexing [43]. A single convolutional layer was implemented with four 3 × 3 kernels followed by a digital FCL, providing a classification accuracy of 95% for the Modified National Institute of Standards and Technology (MNIST) dataset [62]. A frequency comb provides multiple wavelengths that are used to encode the patches, which are subsequently inserted into a photonic computational core. The core implements the corresponding multiplications with 9 × 4 unit-based phase change materials (PCM), arranged in a crossbar topology. Each unit implements a tunable yet nonvolatile synaptic weight [63]. Each output of the crossbar array contains the element-wise multiplication product between the kernel and the patch, whereas the summation is achieved by a photodetector. In the experiment, parallel processing was achieved by utilizing additional wavelengths generated by a frequency comb to simultaneously insert the 4 patches into the PIC. Parallel processing increased the number of required wavelengths from 9 to 36 in this case, along with the number of modulators and detectors needed at the front and back ends, respectively, that is, 36 modulators and 16 analog-to-digital converters (ADCs). At a modulation rate of 14 GHz, this PIC achieved a processing speed of 2 TMAC/s and computed densities of up to 555 GMAC/s/mm^2 with 5-bit precision. For comparison, Google's tensor processing unit (TPU) has a computational density of up to 150 GMAC/s/mm^2 and 8-bit precision [64]. A similar type of network, based on a "broadcast and weight" architecture [58,65], has been proposed with numerical simulations showing 7–14 times faster processing compared to GPUs, while consuming 60%–25% less power, respectively [66].

An alternative multi-wavelength CNN combines spatial, temporal, and spectral encoding strategies to boost performance [67]. A frequency comb was also used, but in contrast to [43,66], wavelengths were used to encode the kernel weights instead of patches. Subsequently, these wavelengths passed through an electro-optic modulator, where each patch was imprinted vector-wise on them. The resulting signal was inserted into a dispersive fiber, which induced a time delay between adjacent wavelengths equal to the modulation rate. At the end of the fiber, a photodiode (PD) accumulated the received multicolor signal, which, owing to dispersion, arrived at the output as an element-wise

product between the patch and kernel. Multiple kernels could be implemented by using additional wavelengths. The collected feature maps were vectorized and inserted into an optical perceptron using the same dispersion-based mechanism. Three 5×5 kernels were used for the MNIST dataset, which required 72 wavelengths. The stride used in this case was 5, and the feature maps produced after passing through an electrical pooling stage were vectorized and inserted into an optical FCL, which was implemented using the same setup [68]. The achieved accuracy was equal to 89.6% and the total computational speed of the convolutional stage was 4.7 TMAC/s. This setup required the same number of modulators and ADCs as in [41]. Although an optical fiber was used as the dispersive medium in this case, photonic crystals or chirped Bragg gratings can be used to miniaturize the scheme [67].

Finally, with respect to coherent meshes, a photonic CNN based on Mach–Zehnder interferometers (MZIs) was theoretically proposed [69]. A matrix containing kernel weights was implemented using an MZI mesh [42]. A single laser injected the inputs as time traces into the mesh. This setup does not exploit multiplexing, and as a result, its computational density is inferior to that of incoherent methods. The power efficiency is similar to that of multi-wavelength PICs because single-wavelength operation leads to fewer power-hungry electronic components at the output. However, an interesting implication in the case of MZI meshes is that, contrary to incoherent schemes, they can, in principle, support on-chip training through backpropagation [41].

In addition to PICs, another major class of notable photonic CNNs includes implementations based on free-space optics [70]. An optical input propagating through a suitable lens configuration can be mapped from the space domain to the Fourier domain and vice versa. If the input is transferred to the Fourier domain, it can be spatially modulated according to a kernel, followed by an inverse Fourier transform at the output, thereby deriving a convolution process. Experimental results have been demonstrated, accomplishing up to 4 peta-operations/s throughput with classification accuracies for the MNIST and CIFAR-10 datasets equal to 98% and 54%, respectively [71]. Although free-space implementations can generate more operations than PICs, they are bulky by design, rendering them inappropriate for applications that demand a low physical footprint, such as edge computing. Moreover, spatial modulators exhibit low modulation rates in the range of several hertz [72] to a few kilohertz [71], thus substantially limiting the reconfiguration speed of these CNNs.

In each of these studies, the primary objective was to directly transfer the CNN operation pipeline to a photonic platform, thereby substituting digital operations with discrete analog computations. In our study, we aim to explore a different route in which convolution is "replaced" by simple multiplication in the spectral domain. Convolution can be accomplished by propagating a time-dependent signal through conventional optical filters with specific transfer functions. In particular, recently we have proposed an alternative approach where multiple optical filters "slice" the optical spectrum [39], thus applying optical kernels of complex weights. Because an optical filter corresponds to a single kernel, the overall physical footprint is significantly reduced. By exploiting the free spectral range of the filters, this scheme was fully compatible with WDM techniques. Therefore, multiple data streams loaded at different wavelengths can be processed simultaneously using the same filter kernel. This concept is further discussed in the following section.

Unconventional Convolutional Processing Based on Optical Spectrum Slicing

Presentation of the concept

In this section, we explore a new proposition for a passive all-optical photonic convolutional accelerator that relies on an OSS technique that enables multiple convolutional kernels in the analog domain. The proposed scheme acts as an unconventional engine that exploits the convolutional operations of parallel bandpass optical filters to extract different spectrotemporal features from the input tensors (images), thus constituting a scalable photonic system with minimal complexity. In contrast to state-of-the-art photonic CNN implementations that are characterized by scalability limitations [66,68,69], significant image preprocessing requirements [43,69], additional control circuitry [43,66,69], and sophisticated setups [43,66,67], the OSS-CNN accelerator offers a promising alternative. The proposed approach exhibits negligible power consumption only attributed to the photodetection and signal modulation stages, operates with zero latency, and demands lightweight image preprocessing, as will be discussed in the following paragraphs.

A recurrent OSS topology was first introduced in [39], in which passive OSS filters equipped with a feedback loop were exploited for the equalization of high-baud-rate optical signals, outperforming state-of-the-art digital processing approaches. We modified this architecture and used each filter as a convolutional kernel. Thus, sophisticated but cumbersome architectures, which aim to reproduce in the photonic domain what is implemented with the use of digital algorithms, were replaced. A widely used classification task based on the MNIST handwriting dataset was used as a benchmark test. The architecture of the OSS-based CNN is illustrated in Fig. 4. It consists of an analog photonic accelerator that implements the convolutional, non-linear activation, and pooling operations of the OSS accelerator and a digital back-end that maps the OSS outputs to MNIST image classes using an FCL. It should be stressed that typical training in the form of backpropagation can be performed only at the digital front-end stage. The parameters of the photonic back-end that perform CNN acceleration can be trained to some extent with the proper configuration of each filter property. A simple filter can be tuned in terms of its central frequency, which affects the kernel's properties. The bandwidth or order of the filter can also be modified if a photonic accelerator is developed on a programmable chip [57].

First, each image is divided into non-overlapping $n \times n$ patches that are serialized and combined into a single uncompressed vector containing all initial image pixel values flattened with a well-chosen orientation to reduce the temporal distance between pixels that are closer in the spatial domain (see Fig. 5). This arrangement requires lightweight preprocessing of data (rearranging the incoming image pixel values), which enhances the dimensionality reduction capabilities of the OSS accelerator. An optical modulator sequentially imprints the image vector values onto the amplitude of a continuous wave optical carrier. The optical power is then equally split among the OSS nodes. Each OSS node is a bandpass optical filter tuned at a different

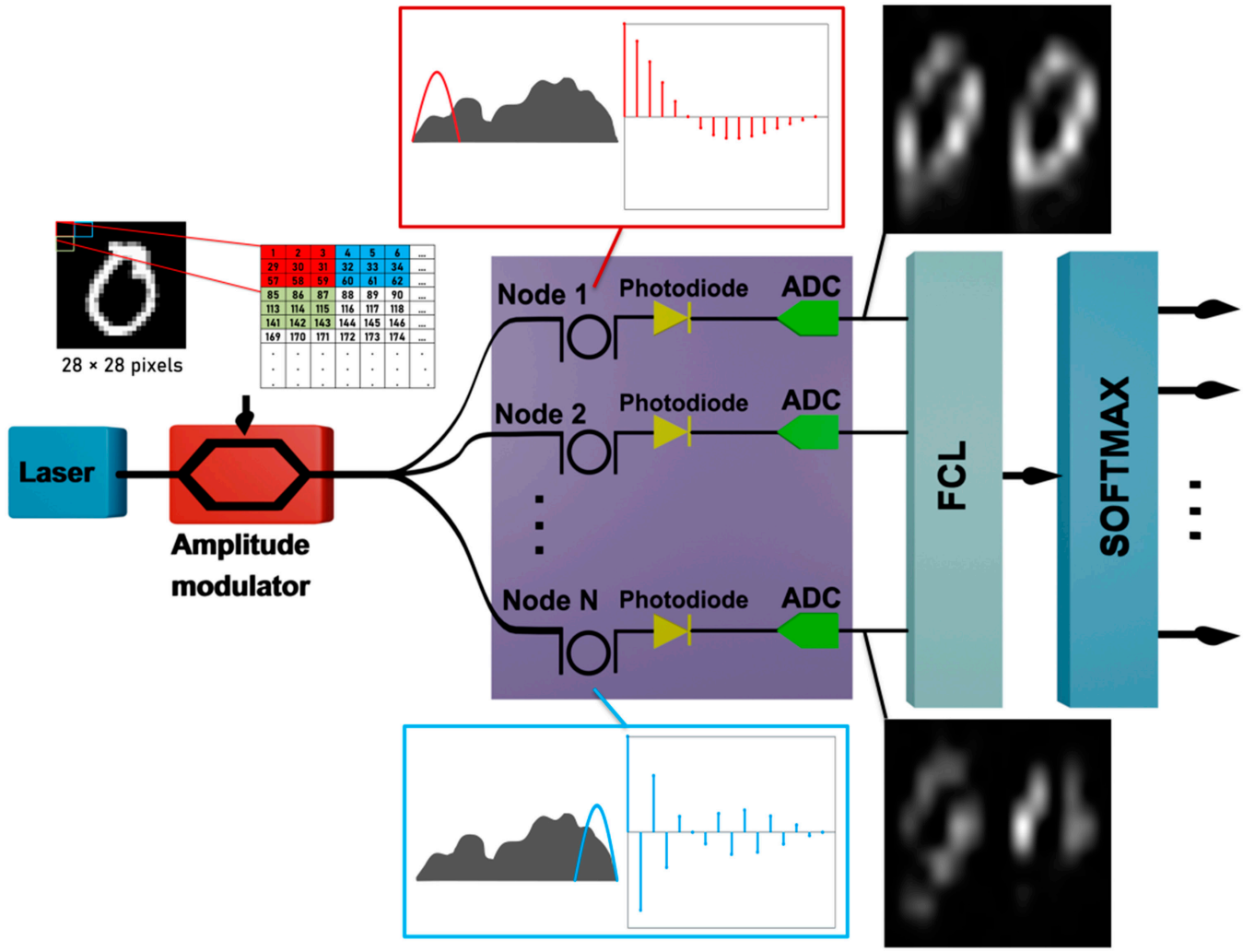

Fig. 4. Conceptual architecture of OSS-CNN: At the input, the image is transformed to a vector that superposes the pixels in the optical temporal domain with the use of an amplitude modulator. The signal is then inserted into the photonic chip, which consists of multiple optical bandpass filters. Each of them slices a specific portion of the input spectrum. Photodetection performs dimensionality reduction through average pooling per slice, and the ADC sends digital data to the FCL-softmax layers for the completion of classification.

central frequency to interact with different spectral components of the optical signal, and therefore extract different features. The transfer function of each filter performs equivalent temporal convolutions governed by its impulse response, which is defined by its order, bandwidth, and central frequency. Thus, each OSS node applies a different set of analog weights to each image patch. This operation is equivalent to the application of different kernels over a certain "receptive field" for each image in traditional CNNs. The differentiated time traces produced at the output of each node were eventually detected using a limited-bandwidth PD. The PD has a twofold mission: it implements a non-linear operation and simultaneously averages the convolved data. First, an element-wise squaring is performed by the PD at the OSS node output acting as a non-linear activation function on the "feature mapped" time traces, with effects similar to those of a ReLU function [73]. If the bandwidth of each PD is set to be inversely proportional to the employed patch size (n^2), the passage of the signal through the PD results in the extraction of an average value over a time slot corresponding to n^2 initial pixels. In this context, an analog averaging operation was performed by mapping the initial image patches to a single value. This is equivalent to the average-pooling layer of a traditional CNN. Eventually, an ADC was placed after each PD to convert the bandwidth-limited analog time series into a sequence of digital samples. The digitized outputs of each node were serialized and fed into a software-based front-end comprising an FCL for classification completion.

Results and discussion

To assess the performance of the accelerator in terms of classification accuracy, throughput, and power consumption, we numerically investigated the impact of the critical hyperparameters on its performance. The hyperparameters of interest are the number of OSS nodes (N), patch mapping size ($n \times n$), input power of the optical signal, OSS filter-node properties, and sampling rate of the ADCs. The investigation was conducted using the MNIST dataset of handwritten digits. In this study, we employed the full MNIST image database, which consists of 60,000 images for training and 10,000 images for testing. Each image was vectorized according to an $n \times n$-pixel patch arrangement, with n taking values ranging from 2 to 5. As shown in Fig. 5, for a 3×3 patching scheme, the pixel values

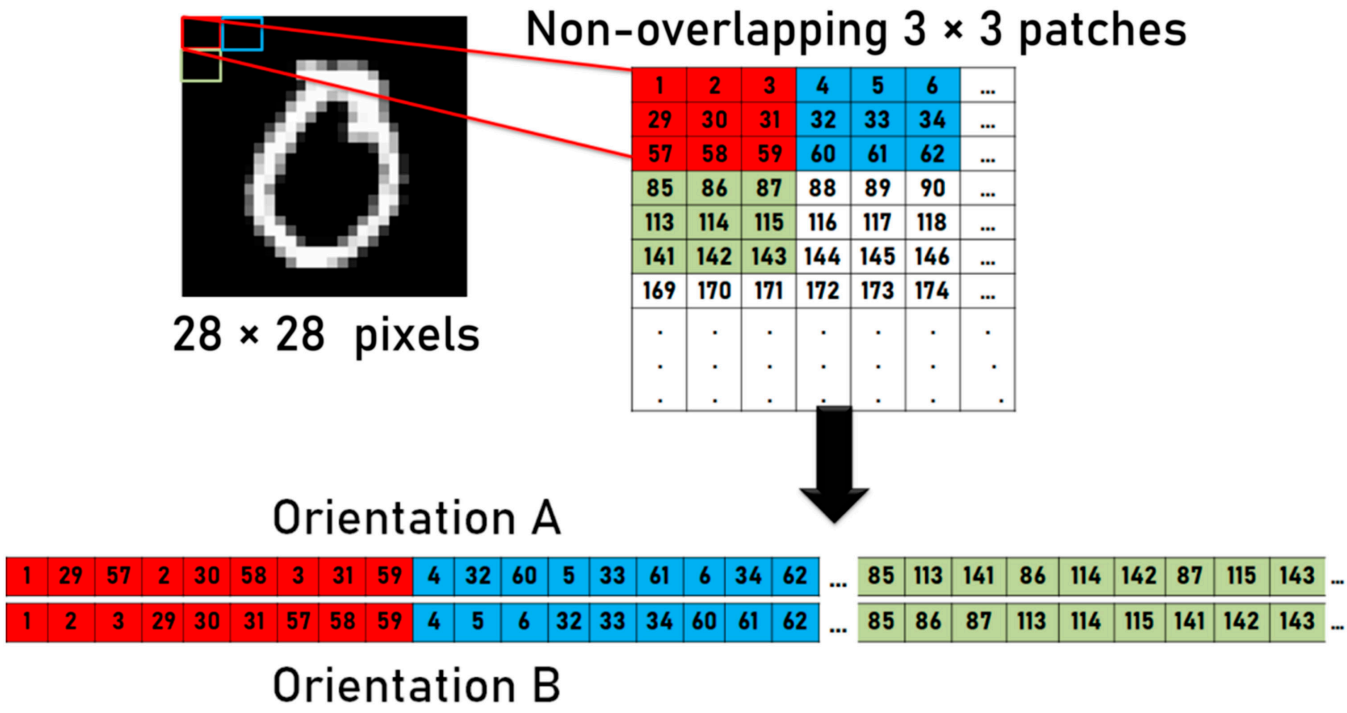

Fig. 5. Division into 3 × 3 patches and serialization of the MNIST image pixel values. The 2 orientations A, B result in 2 vectors per image, and each of them is launched sequentially into each OSS filter to trigger diversified features at the output of the optical filters.

of each patch can be serialized with appropriate orientations to enhance the temporal correlations among adjacent pixels. Two basic orientations were adopted in this study (A and B, as depicted in Fig. 5), which resulted in data augmentation at the input and diversified features at the output of each OSS node. Hence, each MNIST image was flattened to an uncompressed vector twice the initial image size to accommodate the 2 adopted orientations. The padding of zero values was applied around the edges of each image for the 3×3 and 5×5 patches. A digital-to-analog converter (DAC) was used to convert the MNIST data into an analog waveform. This waveform drives the simulated Mach–Zehnder modulator to imprint the input vector values onto the amplitude of the optical carrier at 1,550 nm. The optical signal was split into N OSS nodes using a $1 : N$ coupler. It should be noted here that because passive components are used for the convolutional function, the processing rate of the system is capped by state-of-the-art DAC technologies, which currently marginally surpass 100 Gsa/s [74]. In contrast, the modulation rate can scale up to 500 GHz on plasmonic platforms [75], whereas photodetection or ADC bandwidth is not an issue here because our technique intrinsically reduces the dimensions at the output, thus scaling down the bandwidth requirements by a factor of n^2 with respect to the DAC bandwidth.

OSS nodal filters can be implemented using add/drop microring resonators (MRRs), Mach–Zehnder delay interferometers, or more complex architectures. In this study, we numerically demonstrated that even a first-order filter can perform as an excellent analog convolutional kernel. In principle, the frequency response of a first-order bandpass optical filter is described by a transfer function given by

$$H(f) = \frac{1}{1 + \frac{j(f - f_m)}{f_c}} \quad (1)$$

where f_m and f_c correspond to the central and cutoff frequencies of the bandpass filter, respectively. This filter exhibits an impulse response equal to

$$h(t) = 2\pi f_c e^{-t(2\pi f_c - j2\pi f_m)}, \quad t > 0 \quad (2)$$

Based on Eq. 2, first-order bandpass filters can provide diversified complex-valued kernels in the temporal domain, depending on the bandwidth (f_c) and detuning (f_m) of their central frequency with respect to the carrier frequency of the optical signal. These operations between the optical carrier and OSS nodes are performed instantaneously. In Fig. 6, the impulse responses of 2 arbitrary nodes are depicted, with each of them having $f_c = 3.2$ GHz and detuning of $f_{m1} = 3.2$ GHz and $f_{m2} = 60.8$ GHz with respect to the optical carrier's central frequency. It was assumed that the pixels were fueled by the CNN accelerator at a rate of 128 GSa/s. The depicted impulse responses extend to a 16-pixel time duration and clearly show that they correlate adjacent pixels with an exponentially decaying fading memory depending on f_c. They also exhibit their diversity due to their different f_m values affecting the real part of $h(t)$ (Fig. 6A and B) and its complex representation (Fig. 6C and D). The order and bandwidth of the filters also determine the receptive fields of the OSS node kernels. Increasing the bandwidth f_c of the bandpass filter corresponds to a faster exponential decay of the impulse response, which determines the number of temporal values/pixels that participate in each convolutional operation. Therefore, filters with a narrower bandwidth exhibit a larger temporal memory and involve a wider range of pixels for each convolution, which is analogous to a kernel with a larger receptive field. The order of the filter defines the asymptotic decay of the weights within the time interval of interest. The OSS nodes in this numerical analysis were simulated as

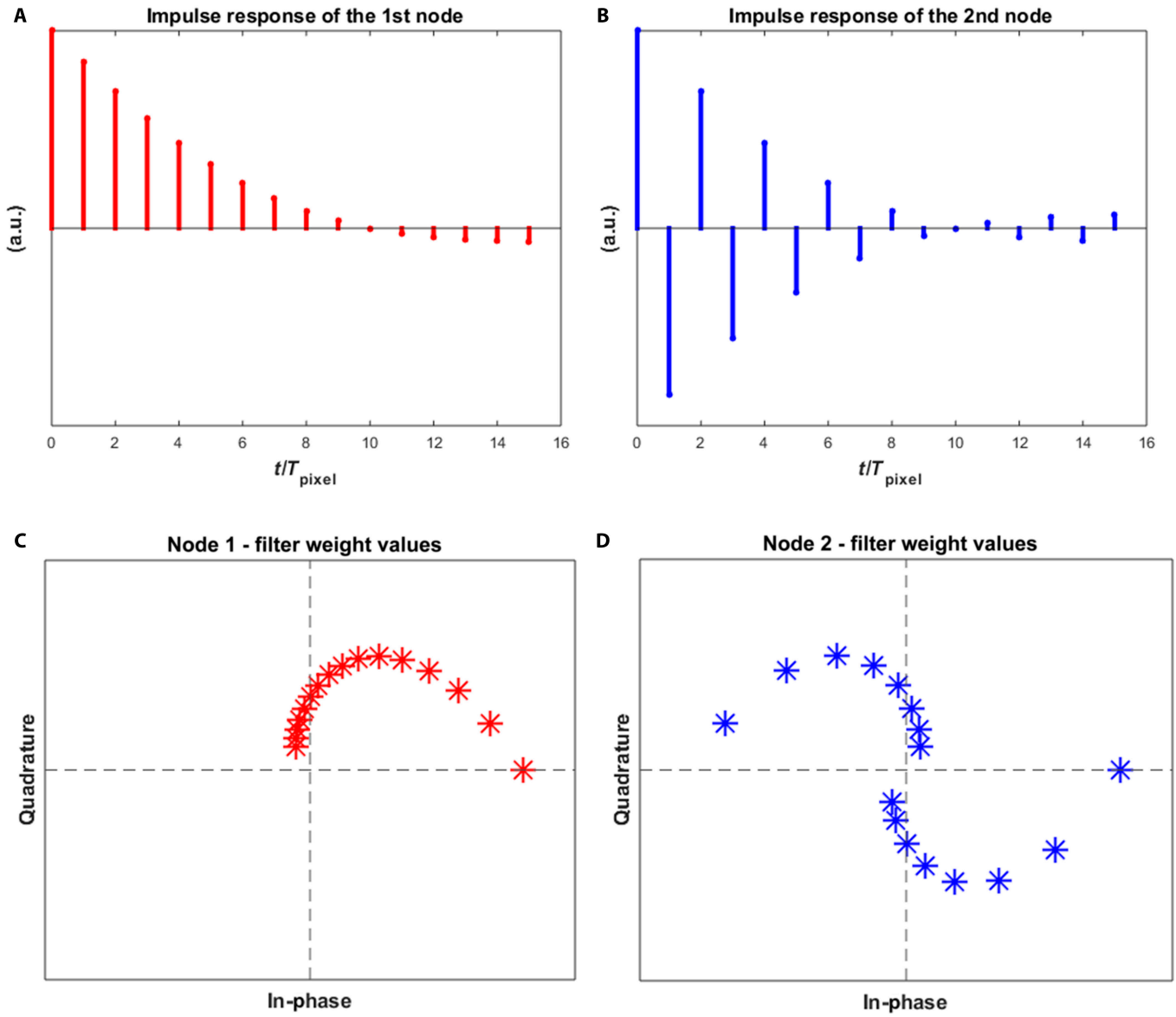

Fig. 6. Real part of the discretized impulse response of 2 detuned OSS nodes with $f_c = 3.2$ GHz and $f_{m1} = 3.2$ GHz (A) and $f_{m2} = 60.8$ GHz (B) illustrated over a 16-pixel time duration. Each filter node applies a different set of analog (complex) weights at the input signal and therefore implements a different convolutional kernel on the image patches. The corresponding complex weights are depicted in the scatterplots of (C) and (D).

first-order MRR filters, whose drop ports were used as the output of the node.

The electric field at the output of the OSS nodes is directed to PDs that are simulated as square-law detectors affected by shot and thermal noise, followed by fourth-order Butterworth filters with a 3-dB bandwidth inversely proportional to the employed patch size given by

$$BW_{PD} = \frac{PR}{n^2} \tag{3}$$

Here, PR is the pixel rate of the input signal and n^2 is the number of total patch elements (patch size). An ADC with a fixed 8-bit precision was simulated after each PD with a sampling rate (SR) that nominally should be set to $SR = 2BW_{PD}$ based on Nyquist; however, in this investigation, it varied from $BW_{PD}/5$ to $2BW_{PD}$ to control the degree of data compression. The ADC output is a digital sequence of averaged convolved data.

The effect of the number of OSS-CNN nodes was investigated for $N = 2, 3, 5, 10$,where N denotes the number of MRR filters. The bandwidth and central frequencies of the nodal bandpass filters were appropriately chosen to fully cover the spectrum of the input signal with a small overlap between adjacent filters. For instance, by employing 2 OSS nodes to fully cover the 64-GHz electrical bandwidth of the input signal, the cutoff frequencies of the filters were set at 16 GHz, while their central frequencies corresponded to 16 and 48 GHz. On the one hand, classification accuracy evidently improved with the use of more filters (more kernels) with a narrower bandwidth, reaching a maximum of 97.6% for 10 OSS nodes. Employing more than 10 filter nodes provided no additional improvements. On the other hand, using fewer filters was beneficial in terms of the required minimum input power, given that the

minimum power was required at each node to overcome the shot and thermal noise levels of the PD. Moreover, the fewer filters were used, the higher the dimensionality of the reduction achieved, which is expressed as the number of inputs of the FCL. The digitized outputs were merged and fed into the FCL. Table 1 summarizes how the inference accuracy is affected by all parameters involved.

In Fig. 7, the testing accuracy of a 5-node OSS-CNN with a 4×4 patch size is plotted against the mean input power of each OSS node. It can be seen that the power above which the proposed accelerator performs stably is −10 dBm/node. Hence, only 500 μW is required for a 5-node scheme and 1 mW is required for a 10-node scheme if no other losses are considered in the chip. These values correspond to a photodetector bandwidth of 8 GHz.

A useful metric for assessing the efficacy of the OSS-CNN in parameter reduction is the compression ratio, defined as the ratio of the uncompressed data (28×28 per image in the MNIST case) to the total digital samples delivered to the input of the FCL (FCL input size) per image. In Fig. 8, the inference accuracy of a 5-node OSS-CNN is plotted against the compression ratio, which depicts the impact of the patch size. The accuracy values of a standalone FCL fed with raw MNIST image pixel values are also presented. A standalone FCL fed with full-size MNIST images achieved accuracy on the order of 92%. It is obvious that an FCL assisted by an OSS-CNN accelerator outperforms a standalone FCL, even when the compression ratio is 4. The 4×4 patch arrangement served as the best solution for this classification task in all OSS-CNN cases, resulting in a maximum accuracy of 97.6% with a 10-node OSS accelerator and a compression ratio of 0.8, which means that a slight increase in the inputs of FCL (from 784 to 980) enhanced accuracy by 5.5%.

Finally, the OSS-CNN accelerator, followed by FCL, was benchmarked against a digitally implemented LeNet-5 architecture on the MNIST image classification task. As detailed in the Convolutional Neural Networks—Background section, the LeNet-5 architecture comprises 7 layers, featuring 3 convolutional layers, 2 subsampling layers, and 2 FCLs (as illustrated in Fig. 2). First, the input layer applied zero padding to the MNIST images, resulting in an input size of 32×32 for the images. The first convolutional layer generates 6 feature maps by utilizing 6 kernels of size 5×5 and employing a ReLU activation function, followed by an average-pooling layer to reduce the feature map dimensions by half to 14×14. The second convolutional layer employs 16 kernels of size 5×5, which again uses a ReLU activation function, resulting in 16 feature maps of size 10×10, followed by a second average-pooling layer that again halves their dimensions. The third convolutional layer comprises 120 kernels of size 5×5 with ReLU activation, generating 120 feature maps of size 1×1. Subsequently, the subsampled outputs were flattened and fed through 2 consecutive FCLs with 120 and 180 neurons equipped with ReLU activation functions. Finally, the output layer includes a 10-neuron softmax layer that produced the probability of a specific input belonging to a particular class. The number of employed FCLs, classification accuracy, and total floating-point operations per second (FLOPS) performed in the testing stage for the MNIST dataset are presented in Table 2 for the 2 different FCL implementations.

It is evident that the performance of both schemes using either conventional CNNs or OSS-CNN as front ends is boosted by the use of a deeper fully connected network. The OSS accelerator, followed by 2 FCLs, was able to achieve an accuracy of 98.2%, slightly worse than that offered by LeNet-5. However, it simultaneously compressed the required FLOPS by almost a factor of 5 at the testing stage. In the case of a single FCL layer, LeNet outperformed OSS-CNN by 1.28% in accuracy, requiring, however, 50 times more FLOPS than OSS-CNN.

The proposed accelerator performs the entire convolutional stage in the analog domain; thus, metrics such as the

Table 1. Testing accuracy as a function of diverse parameters (number of OSS nodes, f_c f_m, patch size, BW_{PD}, SR, and FCL inputs).

OSS nodes	fc (GHz)	fm (GHz)	Patch	BW_{PD} (GHz)	SR (GSa/s)	FCL inputs	Accuracy (%)
		Output Layer = Single FCL – Softmax layer				196	86.02
						392	91.81
						784	92.13
2	16	16, 48	4 × 4	8	8	196	93.95
					16	392	95.15
3	10.66	10.66, 32, 53.33	4 × 4	8	5.33	196	92.53
					16	588	96
5	6.4	6.4, 19.2, 32, 44.8, 57.6	4 × 4	8	3.2	196	94.16
					6.4	392	96.08
					12.8	784	96.76
					16	980	96.85
10	3.2	3.2, 9.6, 16, 22.4, 28.8, 35.2, 41.6, 48.0, 54.4, 60.8	4 × 4	8	1.6	196	93.46
					3.2	392	96.07
					6.4	784	97.27
					8	980	97.6

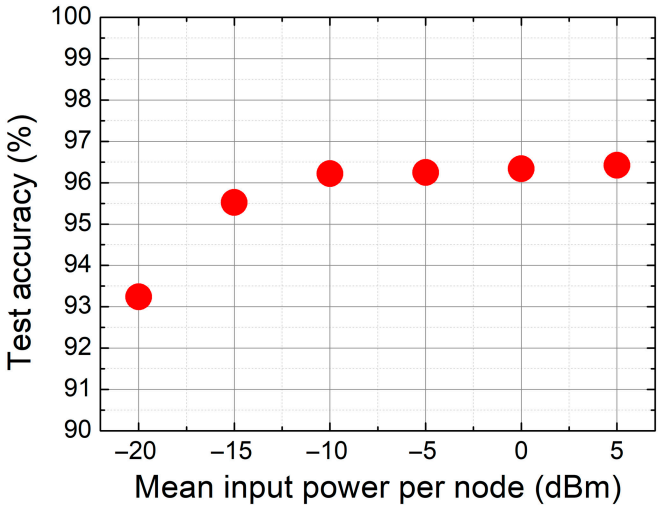

Fig. 7. Testing accuracy as a function of the mean input power per node for a 5-node OSS-CNN with a 4 × 4 patch size and an 8-GHz bandwidth PD.

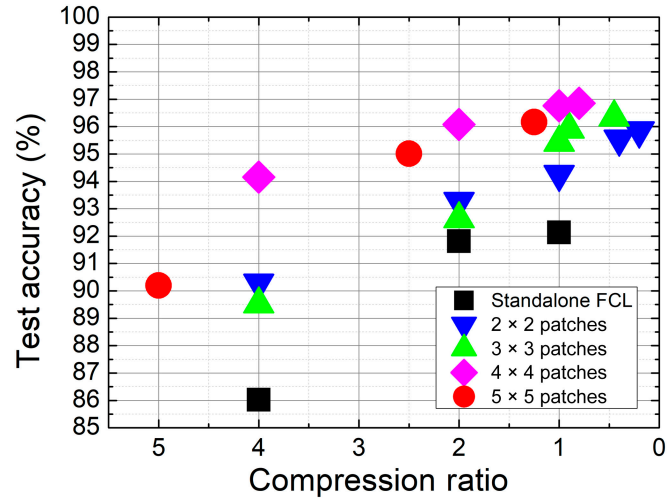

Fig. 8. Comparison of the testing accuracy of a 5-node OSS-CNN with different patch sizes as a function of compression ratio, against a standalone FCL fed with compressed pixel data.

computation speed, power efficiency per computation, and computation density [34] can be approximated by considering a digital equivalent. If we assume that the continuous-time convolution is somehow equivalent to a discretized convolution with a time step dictated by the pixel time, then we can extract its computational efficiency as if it were operating in discrete time. Considering an optical filter with a memory capacity equal to M pixels determined by its bandwidth, it is obvious that for each pixel inserted in the filter, M MAC operations are carried out. Therefore, for N nodes, the input pixel rate is equal to PR and W wavelengths, and the computational speed (number of MACs per second) of the OSS-CNN is equal to $W \times M \times N \times PR$. For a DAC of 128 GSa/s [74], 10 nodes, a single wavelength, and a memory capacity of 16, matching the receptive field of a 4 × 4 kernel, the total setup performs approximately 20.5 TMAC/s. Considering an MRR of a 108-μm radius, with a free spectral range (FSR) that matches the PR, and a distance $\Delta h = 10$ μm between the nodes, the total physical footprint of the OSS filters is approximately equal to $(2.2 \times 2 \times R) \times N \times (2.2 \times 2 \times R + \Delta h) = 2.32$ mm^2. Thus, for a photonic chip containing 10 MRR nodes, the computational density was at least 8.82 TMAC/s/mm^2. In terms of power efficiency, the total power consumption [34,43] is as follows:

$$P = N\left[\frac{h\nu}{\eta} max\left(2^{2N_b+1}, \frac{C_d V_r}{e}\right)\frac{PR}{n^2} + \frac{E_{mod} N_b PR}{N} + E_{ADC} N_b \frac{PR}{n^2}\right] W \quad (4)$$

Here, $\eta = \eta_L \eta_{MRR} \eta_{PD}$ is the combined total quantum efficiency, where η_L is the quantum efficiency of the laser, η_{MRR} is the efficiency of the MRR, and η_{PD} is the efficiency of the photodetector. In addition, $h\nu$ is the energy of a single photon, N_b is the required bit precision, C_d is the capacitance of the PD, V_r is the driving voltage of the PD, e is the charge of the electron, E_{mod} is the energy per bit of the modulator, and E_{ADC} is the energy per bit of the ADC. It is assumed that $C_d = 2.4$ fF, $V_r = 1$V, and $n_L = n_{PD} = 0.1$, whereas for $R = 108$ μm, losses are equal to −0.4 dB/cm and the coupling ratio is equal to 0.1, and the losses at the drop port are −3.5 dB, corresponding to an efficiency of $\eta_{MRR} = 0.45$. Assuming a bit precision equal to $N_b = 5$, $E_{mod} = 1$ pJ/bit [61], $E_{ADC} = 2$ pJ/bit [61], 128 GSa/s modulation rate, and 10 nodes, with 4 × 4 patches and a single wavelength, the total power consumption for a carrier at 1,550 nm is approximately 1.47 W. Therefore, because this setup operates at 20.5 TMAC/s, the computational efficiency is equal to 72 fJ/MAC for 5-bit precision and 127 fJ/MAC for 8-bit precision. For comparison, the multi-wavelength photonic core presented in [43] achieves, for a 64 × 64 optical mesh with the same front-end and back-end parameters and for a single-wavelength channel, a power efficiency equal to 426 fJ/MAC for 5-bit precision and 2.05 pJ/MAC for 8-bit precision, indicating that the OSS-CNN scheme can be advantageous in terms of energy efficiency. In Table 3, we compare state-of-the-art implementations of CNN architectures and show the important advantages of OSS-CNN in terms of power consumption.

Table 2. Image classification scheme, number of FCLs at the front end, testing accuracy, and total floating-point operations per second at the inference stage.

Scheme	Number of FCLs (layer inputs)	Accuracy	FLOPS
LeNet-5	2 (120,10)	>99%	826,000
OSS-CNN	2 (120,10)	98.16%	178,000
LeNet-5	1 (10)	98.88%	736,000
OSS-CNN	1 (10)	97.6%	14,600

It is worth mentioning that although our scheme can be benchmarked with typical CNN accelerators and outperforms them (as shown in Table 3), it provides an alternative route toward CNN acceleration. In particular, approaches such as [43] and other equivalents fully replicate the mathematical operations of a digital CNN, where all weights can be tuned arbitrarily during training. In our case, the CNN complex weights originate from the transfer function of an analog optical filter; thus, they are intertwined and cannot be arbitrarily

Table 3. Comparison between state-of-art digital and photonic CNN architectures with respect to maximum supported clock speed, bit precision, inference accuracy on the MNIST task, energy per MAC, compute density, and sampling rate of the front-end ADC.

CNN architecture	Clock (GHz)	Bit precision	MNIST accuracy (%)	Energy per MAC (pJ/MAC)	Compute density (TMAC/mm^2)	ADC (Gsa/s)
Nvidia Tesla P40 [76]	1.3	8	>99	10.64	0.05	-
Google TPU v4 [77]	1.05	8	>99	1.25	0.2	-
Photonic tensor core [43]	128	5	96.1	1.8 (3 × 3 kernels) 1.32 (4 × 4 kernels)	1.27	128
DEAP [66]	128	6	97.6	2 (3 × 3 kernels) 1.51 (4 × 4 kernels)	-	128
Dispersive processor [67]	128	5	89	1.21 (3 × 3 kernels) 0.68 (4 × 4 kernels)	-	128
OSS-CNN	128	5	97.6	0.18 (3 × 3 kernels) 0.07 (4 × 4 kernels)	4.96 8.82	14.2 8

tuned independently. Therefore, the above benchmark is based on equivalent FLOPs performed if a typical digital CNN implementation is assumed, as OSS-CNN does not perform discrete FLOPs with discrete weights but analog computations with continuous weight values.

Photonic Spiking Convolutional Neural Networks

In the literature, only 2 realizations of spiking CNN photonic accelerators exist; both share the same neuromorphic node implemented by a vertical-cavity surface-emitting laser (VCSEL) under external optical injection [78,79]. The first case focuses on the application of regular-edge detection kernels to images [80]. Although specific handcrafted kernels are used, the kernels of the pretrained software spiking CNN can also be used in the photonic setup [70]. In particular, the kernels perform edge detection in various orientations, namely, vertical, horizontal, and diagonal. The element-wise product between the kernel and patch was implemented in the digital domain, and the results were transferred sequentially to the photonic domain by modulating the strength of the optical injection. Sparsity was demonstrated with respect to the output of the VCSEL-neuron because only a single spike event was produced each time an edge was detected. This setup was used as an accelerator for classification of the MNIST dataset with six 2 × 2 kernels. The photonic spiking convolutional stage was followed by a software spiking CNN implementing 2 additional convolutional stages and an FCL for classification, with the total setup achieving an accuracy equal to 96.1%. Energy efficiency can decrease to 0.4 pJ/spike. For comparison, digital neuromorphic hardware, such as True North, offers an efficiency of 2.5 pJ/spike, indicating the potential of photonic neurons.

The second case used a similar experimental setup to implement a special type of CNNs known as binarized CNNs [81]. In binarized CNNs, each weight is approximated by a binary value, thus increasing the computational speed by 58 times and decreasing the power consumption by 32 times [82]. Such a representation can significantly simplify the digital preprocessing stage that performs the element-wise product. To extract edge features, this scheme requires 4 convolutions using specific binary kernel matrices that focus on different orientations and specialized patches that constitute binary representations of the region around each pixel. This scheme was experimentally validated, and numerical simulations showed the robustness of the spiking scheme, which predicts correct answers even under a low signal-to-noise ratio. The aforementioned operation can also be achieved using a variety of photonic neurons, such as 2-section devices [83–87], optical injection schemes with different active materials [88], microdisk lasers [89], photonic crystals [90], PCM-based neuromorphic nodes [91], and graphene-enhanced silicon MRRs [92].

Photonic neural networks and SNNs share a common challenge: the implementation of a backpropagation algorithm owing to the required computation of gradients [23,70]. Identifying hardware-friendly processes that differentiate analog activation functions in photonics is not a trivial task. Even when derivatives are computed in the electronic domain, the sequential order in which backpropagation is performed complicates the required setup [93]. Although there are additional training strategies besides backpropagation, such as gradient approximation [42], genetic algorithms, and swarm particle optimization [57], they require a considerably higher number of computations and, as a result, do not scale well with the number of trainable parameters. In SNNs, the adoption of backpropagation is a challenging procedure, which has led researchers to consider alternative biologically plausible schemes, such as STDP. As previously stated, this algorithm does not require the derivation of gradients but only tracks the temporal distance between the pre-synaptic and post-synaptic spike events. The training procedure for computing the derivatives can also be shared by photonic spiking networks that encompass the STDP algorithm. As a result, there

are numerous proposals for STDP photonic implementations in the literature that can allow on-chip training of photonic spiking CNNs.

The STDP rule has been demonstrated mainly in photonic platforms containing active materials, such as cross-gain modulation in SOAs [94] and vertical-cavity SOAs [95]. These platforms split the pre-synaptic and post-synaptic spikes of the pump and probe signals injected into the 2 SOAs. The 2 spikes were encoded at different wavelengths. In the first SOA, the timing between the post-synaptic and pre-synaptic spikes was determined by inspecting the amplitude of a probe pre-synaptic spike. Its power is influenced by the carrier variations originating from a strong pump post-synaptic spike. The second SOA tracked the time difference between the pre-synaptic and post-synaptic spikes in a similar fashion. The derived probe signals determine whether the weight must be increased (potentiation) or decreased (depression). The STDP based on active materials require multiple wavelengths. Power consumption at SOAs can also inhibit scalability and power efficiency [36]. An alternative platform based on passive MRRs has been proposed, in which a single wavelength can be used by exploiting the noninteractive clockwise and counterclockwise propagation modes of the ring cavity [96]. This was achieved by coupling the pump and probe signals in the anticlockwise mode. The pump mode modifies the resonance of the ring cavity by changing the refractive index through Kerr and 2-photon absorption effects. This resulted in a modification of the transmissivity of the probe signal, thus mapping the time difference between the 2 spikes on its amplitude. This suggestion can be scaled well with larger photonic SNNs because they do not require multiple wavelengths or significant power consumption. Another passive implementation of the STDP rule has been demonstrated experimentally in the case of a PCM-based neuromorphic perceptron using a feedback loop connecting the output with PCM weights [91].

In this paper, we present numerical results regarding end-to-end unsupervised training of a deep photonic spiking CNN based on the STDP local learning rule. This study uses 2-section VCSEL devices as neurons, which, apart from high fan-in/fan-out capabilities owing to carrier regeneration and low power consumption, perform both inhibitory and excitatory dynamics through electro-optic conversions [97]. Inhibitory dynamics are crucial in the first bioinspired convolutional stage that emulates the receptive field in ganglion cells to extract spatial features [45,46]. The entire setup is a photonic adaptation of the spiking CNN presented in [24]. A time-multiplexed scheme was introduced in the proposed setup, which exploits the ultrafast dynamics of lasers to mimic multiple spiking neurons. Thus, given a physical photonic platform, the option of sacrificing part of its large computational throughput for implementing more complex neural networks is provided.

Generic Spiking Convolutional Neural Network for Image Classification

In this section, we describe the hardware-friendly spiking CNN proposed by Thorpe et al. [24], where training was a bioinspired unsupervised version of STDP. This implementation-agnostic approach is based on an architecture comprising multiple layers of neurons. All neurons were non-leaky integrate-and-fire neurons. In the first layer, the digital image was converted into a set of spikes using multiple differences in Gaussian filters. Each pixel was processed by a set of ON and OFF neurons that could detect both positive and negative contrasts in the target image. The contrast of each pixel was imprinted on the firing time of each neuron using rank encoding. Consequently, pixels with high contrast fired earlier than pixels with low contrast.

The first (encoding) layer was followed by a set of multiple convolutional and pooling layers that are used to extract features from the incoming images. Each neuron in the convolutional layer had its own synaptic weight, which was used to multiply the corresponding inputs. The values of synaptic weights determined the pattern that a neuron was trained to detect. Consequently, each neuron multiplied the received spikes by the corresponding synaptic weights and accumulated them (MAC equivalent operation). Depending on the outcome, the firing of a neuron determined the correlation between the input and the target patterns. In the convolutional layers, neurons were organized into Neural Maps, each targeting the same pattern but at different locations. As the generated spikes propagated toward deeper layers, more complex features were extracted, which were combinations of simpler patterns extracted from the previous layers.

Each convolutional layer was followed by a pooling layer, whose task was to detect the pattern most highly correlated to the corresponding processing area and propagate it to the next convolutional layer. The pooling operation was completed by propagating only the first incoming spike because it had the highest correlation and ignoring the following ones. In this way, the winner-takes-all mechanism was implemented and simultaneously discarded redundant information. After the last pooling layer, which performed a global pooling operation, the outputs were driven to a support vector machine classifier that determined the neuron output.

Training occurred only on the neurons of the convolutional layers. To start training a specific convolutional layer, all previous convolutional layers must have completed their training. The training algorithm was an unsupervised STDP, which was formulated using the following rule:

$$dw_i = \begin{cases} a^+ w_i\left(1-w_i\right) & dt>0 \\ a^- w_i\left(1-w_i\right) & dt<0 \end{cases} \quad (5)$$

where a^+ is the learning rate for the spikes that arrive earlier than the post-synaptic spike, a^- is the learning rate for the spikes that arrive after the post-synaptic spike, dt is the time difference between the pre-synaptic and post-synaptic spikes, and w_i is the weight of the ith synapse. From Eq. 1, it is clear that dt does not affect the weight update (dw_i), whereas at the same time, the terms w_i and $1 - w_i$ constrain the weights to remain between 0 and 1.

Deep Photonic Spiking Convolutional Neural Networks (Deep Spiking CNN)

Presentation of the concept

As previously mentioned, there have been few photonic attempts to implement spiking CNNs; more importantly, they only transfer a single layer in the optical domain, while avoiding the use of optical training. Recently, we presented a deep (5-layer) time-multiplexed network, in which spiking neurons were realized using 2-section (gain absorber) VCSELs. Our intention was not to emulate a single layer, but to fully adapt the software-based spiking CNN presented above. The main objective of our

network was to extract features from incoming images for successful classification. The network was trained in a purely unsupervised manner via STDP, and its performance was evaluated by classifying four 12 × 12-pixel images that depicted digits 5, 6, 7, and 8.

The photonic spiking CNN comprised 5 layers: a contrast detection layer (CDL), 3 convolutional layers, and a classification layer. In addition, there were synchronization layers among the CDL, convolutional layers, and a classification layer, whose task was to synchronize the output spikes of each layer. The setup is illustrated in Fig. 9. The CDL computed the contrast of each pixel and imprinted its value on the firing time of the corresponding spike (time encoding). The operation and architecture of the CDL partially mimic those of biological retinal ganglion cells (RGC). Every RGC has a receptive field that is divided into 2 separate regions: the central and surrounding areas. The output of the RGC is dictated by the amount of incident light in the 2 areas. In the absence of external stimuli, RGC produces spikes at a constant rate, f_r. When the central area receives an external stimulus (the surrounding area is not illuminated), the firing rate of the RGC increases ($f_C > f_r$). Consequently, the central area has an excitatory effect on RGC. However, when the surrounding area is illuminated (the center area is kept dark), the firing rate of the RGC decreases, and for a high input power, it ceases firing completely ($f_s < f_r < f_c$). Therefore, stimuli that are incident on the surrounding area have an inhibitory effect on RGC. Lastly, in case the light illuminates both areas (center and surround) of the receptive field, the RGC produces spikes with a frequency $f_{CS} \approx f_r$ because the excitatory effect of the center and the inhibitory effect of the surround cancel each other out.

In our spiking CNN, the RGC dynamics were implemented using 10 VCSELs and 2 PDs. The pixels of the images were inserted in the CDL via amplitude-modulated electrical pulses: white pixels were inserted via a radiofrequency pulse with 0.2 mW amplitude, whereas black pixels were inserted via 2 μW amplitude pulses. The receptive area of the RGC was implemented using a set of 9 VCSELs placed in a 3 × 3 layout (Fig. 10A). The center VCSEL (C-VCSEL) simulated the behavior of the excitatory center area, whereas the 8 surrounding VCSELs (S-VCSELs) simulated the inhibitory effect of the surroundings. The outputs of the C-VCSEL and S-VCSELs were monitored using 2 separate PDs. The first PD was driven by the output of the C-VCSEL and recorded excitatory stimuli, whereas the second received input from the 8 S-VCSELs and was responsible for inhibitory stimuli. The signals of the excitatory and inhibitory PDs were weighted and summed before entering the RGC-VCSEL, which encoded the contrast value of the firing time of the spike. In detail, the RGC-VCSEL was biased to fire a spike at 3.9 ns under no external stimuli. To implement excitatory and inhibitory effects, the output of the excitatory PD was weighted with a positive weight (w^+) and that of the inhibitory PD was weighted with a negative one (w^-). Moreover, to fully simulate the effects of the center and surround area for all possible cases, the weight of the excitatory PD must be 8 times higher than that of the inhibitory one ($w^+ = 8w^-$).

The reason for this modification was that the excitatory PD was driven by one VCSEL, whereas the inhibitory PD was driven by 8 VCSELs. The weighted sum of the signals from the PDs drove the RGC-VCSEL and, depending on its sign, dictated the latency of the spike. Specifically, if the weighted sum was positive, the RGC fired earlier, whereas if the weighted sum

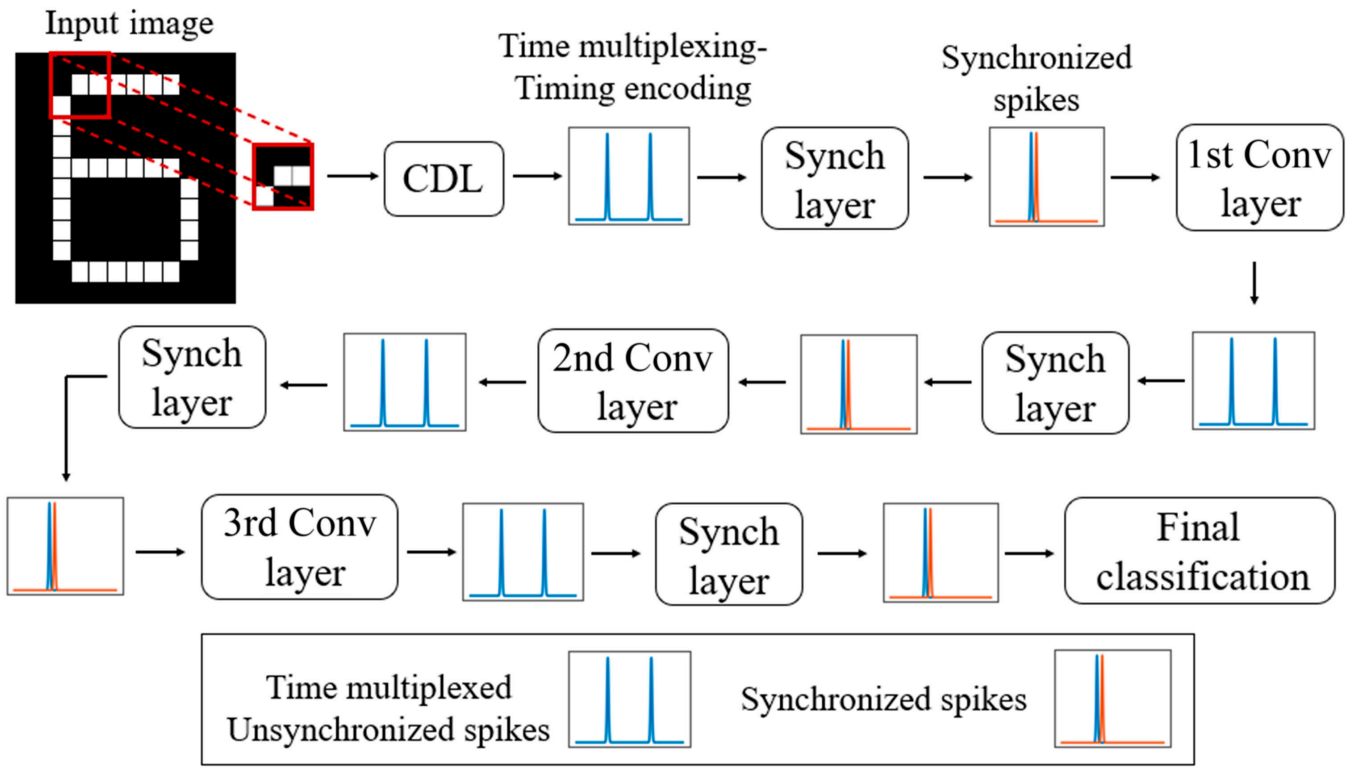

Fig. 9. Architecture of the time-multiplexed convolutional SNN trained with unsupervised STDP. Firstly, the image is processed by the contrast detection layer (CDL) in a serial manner. Then, the spikes are inserted into the synchronization layer that synchronizes the spikes before entering the convolutional layer. This procedure is followed in subsequent layers, and the final data are classified in a fully connected layer.

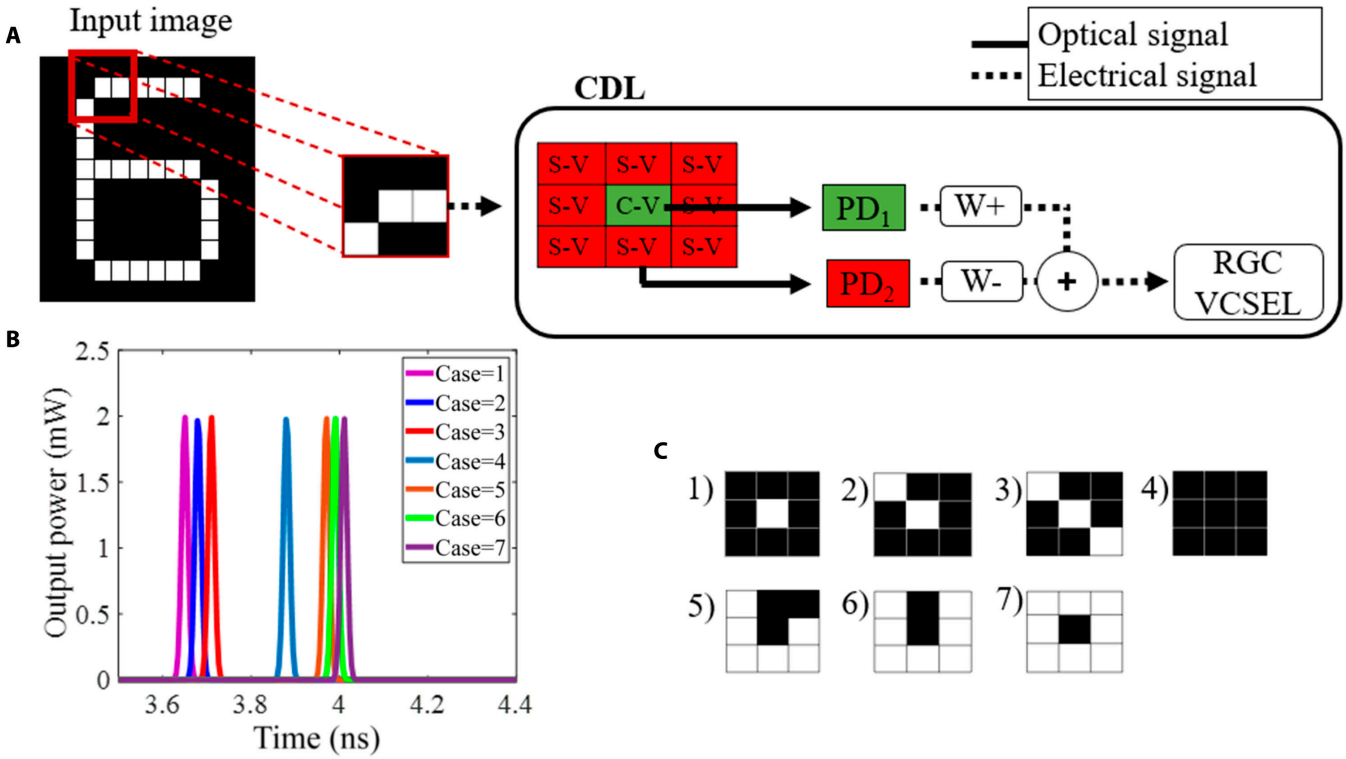

Fig. 10. (A) Architecture of the CDL. Each pixel is sequentially inserted into the CDL. The surround pixels are electrically injected into the S-VCSELs, while the center pixel is inserted into the C-VCSEL. (B) Spiking output of the RGC neuron. Positive contrast (cases 1, 2, and 3) produces fast spikes, while negative contrast (cases 5, 6, and 7) produces spikes with high latency. (C) The input patterns that produce the spikes in the diagram in (B). The C-VCSEL is designated with green color due to its excitatory effect, while S-VCSELs are designated with red color due to their inhibitory effect.

was negative, the RGC fired with an enhanced delay. Figure 10B shows the firing time of the RGC-VCSEL for the various input patterns presented in Fig. 10C.

The CDL is followed by a synchronization layer (Fig. 11), whose purpose is to synchronize and align the incoming pulses in a specific time slot. Specifically, each pixel is processed by the CDL inside a specific time slot with duration $T_{pix} = 5$ ns. Each timeslot starts at $(k-1)T_{pix}$ and ends at kT_{pix} (k denotes the number of processed pixels). Due to the applied time-multiplexed processing, each pixel has its own time reference, which is the beginning of its time slot ($(k-1)T_{pix}$). Different time references would render the application of the STDP algorithm and the convolution operation ineffective, because the spikes that originate from a specific area do not enter the convolutional layer within a common time frame. This would render the computation of the time differences (dt) between the pre-synaptic (t_{pre}) and post-synaptic spikes (t_{post}) useless because they do not express the real relationship between them. Therefore, a synchronization layer is necessary, and its role is to impose an $(m-l)T_{pix}$ delay for the kth spike, where m is the size of the convolutional window (CW) in each layer, and l is the remainder of k divided by m. Using this technique, the spikes of a specific m-pixel area acquire a common time reference that allows the proper use of the STDP algorithm and convolutional processing. In electro-optic synapses, the synchronization layer can be easily implemented using a predetermined static electrical delay line. It is worth mentioning that the Synch Layer is necessary only when the TDM scheme is employed, in which case a single neuron processes multiple patches of the image, thus relaxing the hardware requirements. If TDM is omitted, the number of neurons would increase, but STDP would be implemented directly.

Each convolutional layer comprises multiple neurons (VCSELs) that detect patterns in incoming spike trains. In classical spiking CNNs, a neuron detects a specific pattern in a restricted area of an image. However, a neuron in the photonic spiking CNN detects the same pattern in the entire image using TDM processing. This enables a radical reduction in neuron count, which has a direct impact on the power consumption of the network.

The convolutional layer differs during the training and testing procedures. During training, incoming spikes were inserted into the first neuron of the convolutional layer, and the usual weighted addition was performed. If a neuron fired a spike, the incoming pattern was recognized successfully. This triggers 2 additional procedures. The first is the update of the weights of the first neuron, whereas the second is the transmission of a lateral inhibition signal to all other neurons. This signal lowered the bias of the neurons, driving them outside the spiking regime, rendering spiking impossible. If the first neuron did not recognize the incoming neuron, the spikes were directed to the second neuron with a delay T_D and the same procedure was repeated for all subsequent neurons. Time interval T_D must be at least equal to the time required by a neuron to successfully recognize a pattern. The same procedure was repeated until the pattern was successfully recognized by a neuron. When the convolutional layer completed its training, the T_D delays were omitted along with the lateral inhibition signal, and the testing of the photonic spiking CNN could begin. During the testing

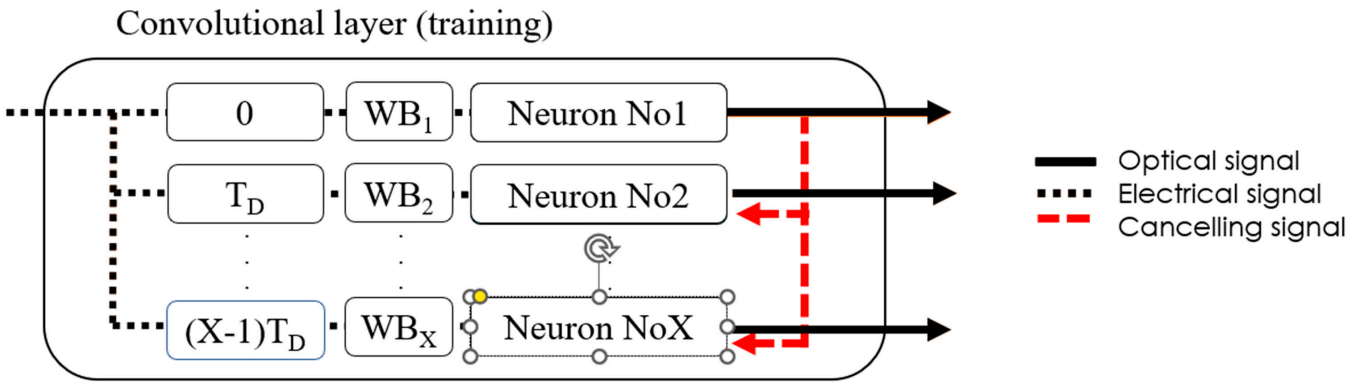

Fig. 11. Architecture of a convolutional layer during the training phase. During inference, the delays T_D and the cancelling signals are omitted.

phase, the incoming synchronized spikes were simultaneously transmitted to all the neurons of the convolutional layer.

With regard to the training algorithm, the first convolutional layer was trained using the classical unsupervised STDP, whereas the second and third convolutional layers were trained using a slightly modified version. In particular, if the corresponding synapses carried a spike, their weight increased by 0.1. If the synapses did not carry a spike, their weight decreased by 0.1. The following learning rules are summarized in the following formula:

$$dw = \begin{cases} 0.1 & \text{spike} \\ -0.1 & \text{no spike} \end{cases} \quad (6)$$

The classification layer is an FCL, as described above, which classifies incoming images based on the extracted features of the previous layers.

Results and discussion

In this section, the numerical results of the training and inference operations of the photonic spiking CNN are presented. The proposed network was trained to classify four 12 × 12-pixel images depicting digits 5 to 8. Consequently, the training set consisted of 4 images repeatedly inserted into the network. It is worth mentioning that no labeling was needed to train the network. First, the first convolutional layer (CL_1) was trained. Its processing area, designated as convolutional window 1 (CW_1), consisted of a 3 × 3-pixel layout. Therefore, the number of inputs to the first layer was 9 (one input for each pixel of CW_1), whereas the stride of CW_1 was equal to one pixel. With this meticulous scanning, CL_1 was able to detect most features and learned 33 patterns (Fig. 12B). When the learning of CL_1 was completed, the training of CL_2 began. The processing area of the second layer (CW_2) consisted of 4 CW_1 or a total area of 6 × 6 pixels. By considering that each CW_1 may match one of the 33 learned patterns, then the total number of inputs for the CL_2 must be 4 • 33 = 132 inputs, where the inputs 1 to 33 corresponded to the first CW_1, the inputs 34 to 66 corresponded to the second CW_1, and so on. Because of the detailed scanning in the training of CL_1, no overlap was required during the training of CL_2. Therefore, the stride for CL_2 was set to 6 pixels. In CL_2, 9 patterns were learned, which were combinations of the simpler patterns learned in CL_1. Finally, when CL_2 completes its training, CL_3 may start training. Following the principles governing the training of CL_1 and CL_2, CL_3 has a scanning window CW_3 that consists of 2 CW_2 or 6 × 12 pixels. The number of inputs was 2 • 9 = 18 and no overlap was required. With this configuration, CL_3 learned 6 patterns, which were combinations of patterns learned by CL_2.

With regard to the inference stage, the proposed photonic spiking CNN accurately classified all 4 digits. To ensure the proper functioning of the network, a different training set consisting of digits 1 to 4 was utilized, resulting in successful training because the network could extract new features and label the input images. To examine the limitations of the network, 2 types of noise were introduced: white additive Gaussian noise, which affects the instant power level of the inserted rectangular pulses, and noise affecting the intensity of the pixels, which subsequently affects the mean power level of the rectangular pulses. The first type of noise had no impact on the network performance because of the integration attribute of the neurons, which negated the effects of instant power changes. However, the second noise type had a detrimental impact as the network was highly susceptible to it, with a standard deviation of 12%, rendering the network incapable of classifying incoming images (Fig. 13A). The primary advantage of the photonic spiking CNN was its ability to tune the neuron count, an ability made possible by the time-multiplexing technique, which resulted in a significant reduction in the number of neurons from 2,020 to only 52 (Fig. 13B). This exponential decrease was because each neuron detected a specific pattern in the entire image. The reduction in the number of actual neurons also had a beneficial impact on power consumption, although it came at the cost of lower processing rates, with image processing latencies ranging from 5 ns for 2,020 neurons to 720 ns for 52 neurons (Fig. 12B). By incorporating multiple layers, including CDLs, synchronization layers, and convolutional layers, the images were processed in parallel, thereby decreasing the processing latency of the network by up to 5 ns, which is the time required for the network to process a single pixel. Consequently, the processing rate of the network can be adjusted according to the specific requirements of the task. This clearly demonstrates the trade-off between the processing rate of the network and the neuron count.

In Table 4., the performance of the proposed network is compared with that of other spiking implementations in terms of neuron count, number of synapses, energy efficiency, addressed task, and type of implementation. The key property of our proposed network is its low energy consumption (0.16 pJ/spike), which originates from 2 main characteristics of the implementation: the low power consumption of the VCSEL-neuron (1 mW per neuron) and the time-multiplexing technique, which radically decreases the actual neuron count. In general, the

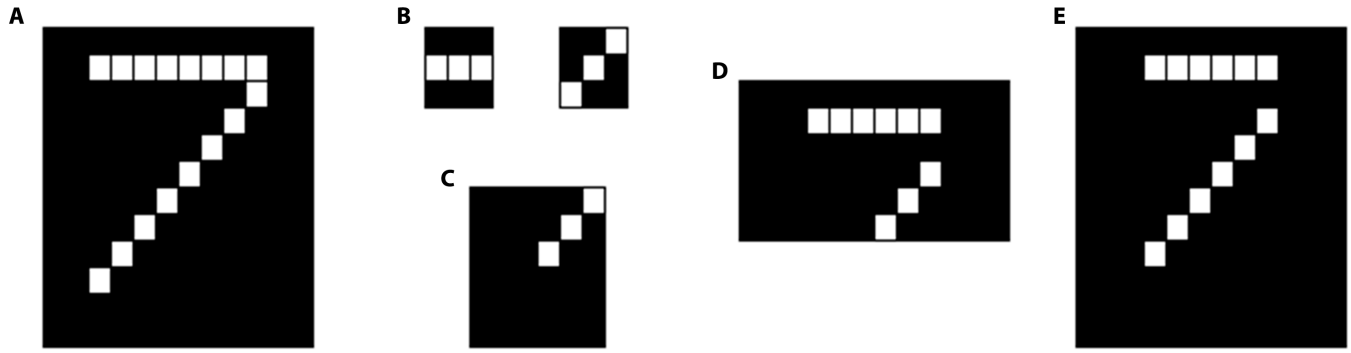

Fig. 12. (A) Training image of digit 7. (B) Features learned in CL1. (C) Features learned in CL2. (D) Features learned in CL3. (E) The abstract version of the digits that the network identifies.

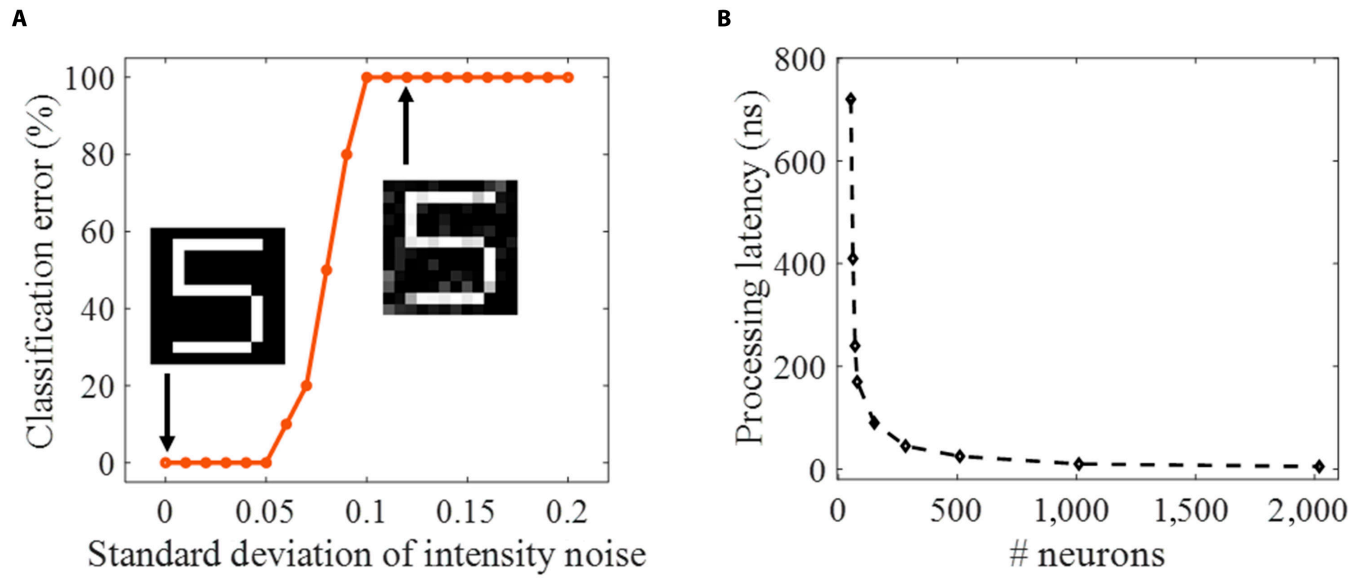

Fig. 13. (A) Performance of the photonic spiking CNN for different standard deviation values of the noise source. The noise is drawn from a normal distribution whose standard deviation is expressed as a percentage of the nominal input power for white pixels ($P_{IN} = 0.2$ mW [x-axis]). In this figure, digit "5" is presented as an indicative example of the intensity noise's impact. The left image of digit "5" represents the case of no intensity noise while the right one corresponds to an intensity noise of 12%. (B) The processing latency of the incoming 12×12-pixel image for different numbers of neurons.

decrease of actual neurons permits the substantial decrease of power consumption as the energy per spike is reduced by a factor that ranges from 2.5 at [80] to 15.26 times [27], whereas its accuracy is similar to other implementations [91,95]. The energy efficiency of the proposed method was evaluated using the following formula [27]:

$$E_p = \frac{P}{fN_{tot}} \tag{7}$$

where E_p is the energy per spike, P is the total power consumption, f is the frequency of the spikes produced, and N_{tot} is the total number of synapses. To estimate E_p, a power consumption of 1 mW per neuron was assumed, which is typical for VCSELs.

Conclusion

This study highlights that the use of photonic technologies in the neuromorphic arena is a critical step toward minimizing power consumption and boosting processing speed in artificial intelligence applications based on CNNs. The fruits of this effort are ripe in terms of J/MAC or MAC/s that have driven a significant part of the photonic community to strive for precise replication of the "algorithmic" neural network in the photonic hardware. Nonetheless, we believe that this effort, although interesting, has 3 generic limitations that could impede its evolution. The first key drawback is scalability. Application-wise, even basic CNNs (i.e., the MNIST dataset) demand matrices of 28×28, a task that is not straightforward to achieve even with MZI architectures, owing to the accumulation of optical losses and, more importantly, the complexity of the electro-optic driving circuits. A deep architecture makes this problem even more challenging. The solution of multiplexing is a route to circumvent this issue; however, it comes with an elevated cost/footprint because it requires sophisticated hardware. Secondly, one of the most prominent features of a neural network is its ability to represent unknown data using training procedures. Unfortunately, in optics, only basic training schemes have been realized (e.g., gradient descent), whereas the holy grail of neural training, backpropagation, still lacks a realistic on-chip implementation. This limits the applicability of photonic circuits for inference,

Table 4. Comparison between state-of-art digital and photonic CNN architectures with respect to number of neurons, number of synapses, power efficiency and addressed task.

Scheme	Neurons	Synapses	Energy efficiency	Task	Type
True North [27]	1 million	256 million	2.5 pJ/spike	Multi-object detection and classification	Electronic
Loihi (Kapoho Bay) [98]	131,072	130 million	2.3 pJ/spike	Interface with Nengo, address event representation, sensor, and robotic platforms	Electronic
VCSEL edge detection [80]	6	4	0.4 pJ/spike	MNIST	Photonic
VCSEL based SNN [95] [a]	410	4,000	4.1 pJ/spike	Classification of artificial images	Photonic
This work [a]	52	1,617	0.16 pJ/spike	Classification of artificial images	Photonic

[a]The energy efficiency is computed according to Ref. [27].

hindering real neural operations. Consequently, digital electronics, and their desired data-storage capabilities, remain irreplaceable for real-life applications. Third, the nature of light itself changes the basic principles of CNNs; that is, instead of analog negative/positive values propagating in the network, we have complex optical values. This changes the basic rules for the network design and can affect the type of non-linear activation function required. Therefore, we believe that strict replication efforts limit the true potential of neuromorphic photonics.

Alternatively, a viable route is the use of photonics as accelerators. PICs either replace a well-chosen part of digital processing or provide an unconventional operation that, although not present in the digital embodiment of CNNs, can provide equivalent functionality accompanied by critical performance enhancement. This enhancement is either an accuracy boost or a parameter reduction with minimal impact on accuracy. In this context, OSS-CNN does not act as a conventional CNN layer, meaning that it provides a convolution of the signal with arbitrary weights, but these weights are not fully regulated, whereas at the same time, spectral decomposition and dimensionality reduction are offered. Although such a layer is not present in a typical CNN pipeline, our results demonstrate a significant improvement through optical preprocessing. This boost is manifested either compared to a simple FCL resulting in both accuracy enhancement and parameter reduction or compared to a full-scale CNN (LeNet-5), where a radical parameter reduction is achieved with a marginal impact on accuracy. This work also highlighted that isomorphism to biological neural networks can also be achieved by using excitable artificial neurons with identical dynamics, synapses with weighting plasticity, etc. In this framework, innovative work has been conducted, starting from a simple 2-neuron system and moving toward a more realistic neural system with a collaborative interaction of hundreds of neurons. In this field, our proposition exceeds the state of the art by demonstrating the first-time simulation results of a full-scale photonic spiking network incorporating neural dynamics, biocompatible STDP training, retina-like preprocessing, and hybrid information encoding (event-based). In this context, we fuse biological efficiency with photonic performance, and the aforementioned scalability problem is partially addressed using TDM schemes.

Our midterm objective was to develop a true hybrid platform that embeds all 3 aspects: photonic acceleration, isomorphic spiking efficiency, and lightweight digital training. Toward this direction, we started designing a reconfigurable silicon photonic platform able to offer "plastic," low-loss synapses that interconnect III–V lasers acting as TDM neurons generating events. Laser neurons receive preprocessed images offered through OSS-CNN schemes that spectrally decompose incoming digital data, whereas the readout layer is maintained in the digital domain to unlock applicability. This vision was accomplished in the context of the Horizon Europe Project PROMETHEUS (https://prometheus-he.eu).

Acknowledgments

Funding: This work received funding from the EU H2020 NEoteRIC project under agreement 871330 and the EU Horizon Europe PROMETHEUS project under grant agreement 101070195. **Author contributions:** A.T. performed the simulations related to OSS-CNN and wrote the manuscript with the help of all authors. G.S. assisted A.T. with the OSS-CNN simulations and performed a major part of the literature review. M.S. conducted numerical studies on spiking CNNs based on VCSELs. S.D. contributed to comparative studies using conventional machine learning CNNs such as LeNet. K.S. contributed to the simulation of OSS-CNN. D.D. and G.T. contributed to the preprocessing of MNIST images. A.B. and C.M. conceived the concepts of OSS-CNN and spiking CNNs and orchestrated the work. **Competing interests:** The authors declare that they have no competing interests.

Data Availability

Data are available upon request.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444.
2. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323(6088):533–536.
3. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278–2324.

4. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Proces Syst*. 2012;25.
5. Zeiler MD, Fergus R. *Visualizing and understanding convolutional networks*. New York: Springer, Cham; 2014.
6. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv. 2014. https://doi.org/10.48550/arXiv.1409.1556
7. Szegedy C, Liu W, Jia Y, Sermanet P, Reed P, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015 Jun 7–12; Boston, MA.
8. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV.
9. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015 Jun 7–12; Boston, MA.
10. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014 Jun 23–28; Columbus, OH.
11. Hinton G, Deng L, Yu D, Dahl G, Mohamed AR, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process Mag*. 2012;29(6):82–97.
12. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. Mastering the game of go with deep neural networks and tree search. *Nature*. 2016;529(7587):484–489.
13. AI and compute. OpenAI. 2018 May 16. [accessed 2022 Jun 2] https://openai.com/blog/ai-and-compute/
14. Li X, Zhang G, Huang HH, Wang Z, Zheng W. Performance analysis of GPU-based convolutional neural networks. Paper presented at: 2016 45th International Conference on Parallel Processing (ICPP); 2016 Aug 16–19; Philadelphia, PA.
15. Jones N. How to stop data centres from gobbling up the world's electricity. *Nature*. 2018;561(7722):163–166.
16. Dayan P, Abbott LF. *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge: MIT Press; 2005.
17. Thorpe S, Delorme A, Van Rullen R. Spike-based strategies for rapid processing. *Neural Netw*. 2001;14(6-7):715–725.
18. Izhikevich EM. *Dynamical systems in neuroscience*. Cambridge: MIT Press; 2007.
19. Jolivet R, Gerstner W. The spike response model: A framework to predict neuronal spike trains. In: Kaynak O, Alpaydin E, Oja E, Xu L, editors. *Artificial neural networks and neural information processing — ICANN/ICONIP 2003. ICANN ICONIP 2003*. Berlin, Heidelberg: Springer; 2003.
20. Pfeiffer M, Pfeil T. Deep learning with spiking neurons: Opportunities and challenges. Frontiers in Neuroscience. 2018 May 22. [accessed 2022 Mar 14] https://www.frontiersin.org/article/10.3389/fnins.2018.00774
21. Diehl PU, Neil D, Binas J, Cook M, Liu S-C, Pfeiffer M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. Paper presented at: 2015 International Joint Conference on Neural Networks (IJCNN); 2015 Jul 12–15; Killarney, Ireland.
22. Song S, Miller KD, Abbott LF. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat Neurosci*. 2000;3(9):919–926.
23. Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A. Deep learning in spiking neural networks. *Neural Netw*. 2019;111:47–63.
24. Kheradpisheh SR, Ganjtabesh M, Thorpe SJ, Masquelier T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw*. 2018;99:56–67.
25. Tavanaei A, Maida AS. Multi-layer unsupervised learning in a spiking convolutional neural network. Paper presented at: 2017 International Joint Conference on Neural Networks (IJCNN); 2017 May; Anchorage, AK.
26. Backus J. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Commun ACM*. 1978;21(8):613–641.
27. Merolla PA, Arthur JV, Alvarez-Icaza R, Cassidy AS, Sawada J, Akopyan F, Jackson BL, Imam N, Guo C, Nakamura Y, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*. 2014;345(6197):668–673.
28. Furber SB, Galluppi F, Temple S, Plana LA. The spinnaker project. *Proc IEEE*. 2014;102(5):652–665.
29. Schemmel J, Briiderle D, Griibl A, Hock M, Meier K, Millner S. A wafer-scale neuromorphic hardware system for large-scale neural modeling. Paper presented at: Proceedings of 2010 IEEE International Symposium on Circuits and Systems; 2010 May 30; Paris, France.
30. Davies M, Srinivasa N, Lin TH, Chinya G, Cao Y, Choday SH, Dimou G, Joshi P, Imam N, Jain S, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*. 2018;38(1):82–99.
31. Shafiee A, Nag A, Muralimanohar N, Balasubramonian R, Strachan JP, Hu M, Williams RS, Srikumar V. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *ACM SIGARCH Comput Architect News*. 2016;44(3):14–26.
32. Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y. Binarized neural networks. *Adv Neural Inf Proces Syst*. 2016;29.
33. Umuroglu Y, Fraser NJ, Gambardella G, Blott M, Leong P, Jahre M, Vissers K, Finn: A framework for fast, scalable binarized neural network inference. Paper presented at: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays; 2017 February; Monterey, CA.
34. Nahmias MA, de Lima TF, Tait AN, Peng H-T, Shastri BJ, Prucnal PR, Photonic multiply-accumulate operations for neural networks. *IEEE J Sel Top Quantum Electron*. 2020;26(1);7701518.
35. Siew SY, Li B, Gao F, Zheng HY, Zhang W, Guo P, Xie SW, Song A, Dong B, Luo LW, et al. Review of silicon photonics technology and platform development. *J Lightwave Technol*. 2021;39(13):4374–4389.
36. Shastri BJ, Tait AN, Ferreira de Lima T, Pernice WHP, Bhaskaran H, Wright CD, Prucnal PR. Photonics for artificial intelligence and neuromorphic computing. *Nat Photonics*. 2021;15:102–114.
37. Tait AN, Ma PY, Ferreira de Lima T, Blow EC, Chang MP, Nahmias MA, Shastri BJ, Prucnal PR. Demonstration of multivariate photonics: Blind dimensionality

reduction with integrated photonics. *J Lightwave Technol*. 2019;37(24):5996–6006.
38. Argyris A, Bueno J, Fischer I. Photonic machine learning implementation for signal recovery in optical communications. *Sci Rep*. 2018;8:8487.
39. Sozos K, Bogris A, Bienstman P, Sarantoglou G, Deligiannidis S, Mesaritakis C. High-speed photonic neuromorphic computing using recurrent optical spectrum slicing neural networks. *Commun Eng*. 2022;1:24.
40. De Lima TF, Peng H-T, Tait AN, Nahmias MA, Miller HB, Shastri BJ, Prucnal PR. Machine learning with neuromorphic photonics. *J Lightwave Technol*. 2019;37(5):1515–1534.
41. Pai S, Sun Z, Hughes TW, Park T, Bartlett B, Williamson IAD, Minkov M, Milanizadeh M, Abebe N, Morichetti F, et al. Experimentally realized in situ backpropagation for deep learning in nanophotonic neural networks. ArXiv. 2022. https://doi.org/10.48550/arXiv.2205.08501
42. Shen Y, Harris NC, Skirlo S, Prabhu M, Baehr-Jones T, Hochberg M, Sun X, Zhao S, Larochelle H, Englund D, et al. Deep learning with coherent nanophotonic circuits. *Nat Photonics*. 2017;11(7):441–446.
43. Feldmann J, Youngblood N, Karpov M, Gehring H, Li X, Stappers M, le Gallo M, Fu X, Lukashchuk A, Raja AS, et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature*. 2021;589(7840):52–58.
44. Prucnal PR, Shastri BJ, Teich MC. *Neuromorphic photonics*. United Kingdom: CRC Press; 2017.
45. Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. *J Physiol*. 1959;148(3):574.
46. Olshausen BA, Field DJ. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*. 1996;381(6583):607–609.
47. Rehn M, Sommer FT. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *J Comput Neurosci*. 2007;22(2):135–146.
48. Bell AJ, Sejnowski TJ. The 'independent components' of natural scenes are edge filters. *Vis Res*. 1997;37(23):3327–3338.
49. Masquelier T, Thorpe SJ. Unsupervised learning of visual features through spike timing dependent plasticity. *PLOS Comput Biol*. 2007;3(2):e31.
50. Panda P, Roy K. Unsupervised regenerative learning of hierarchical features in Spiking Deep Networks for object recognition. In: *2016 International joint conference on neural networks (IJCNN)*. Vancouver (Canada): IEEE; 2016. p. 299–306.
51. Lee JH, Delbruck T, Pfeiffer M. Training deep spiking neural networks using backpropagation. *Front Neurosci*. 2016;10:508.
52. Esser SK, Merolla PA, Arthur JV, Cassidy AS, Appuswamy R, Andreopoulos A, Berg DJ, McKinstry JL, Melano T, Barch DR, et al. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc Natl Acad Sci USA*. 2016;113(41):11441–11446.
53. Rueckauer B, Lungu I-A, Hu Y, Pfeiffer M, Liu S-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front Neurosci*. 2017;11:682.
54. Maass W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw*. 1997;10(9):1659–1671.
55. Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, Shelhamer E. cudnn: Efficient primitives for deep learning. ArXiv. 2014. https://doi.org/10.48550/arXiv.1410.0759
56. Bogaerts W, Pérez D, Capmany J, Miller DAB, Poon J, Englund D, Morichetti F, Melloni A. Programmable photonic circuits. *Nature*. 2020;586(7828):207–216.
57. Pérez-López D, López A, DasMahapatra P, Capmany J. Multipurpose self-configuration of programmable photonic circuits. *Nat Commun*. 2020;11(1):6359.
58. Tait AN, Nahmias MA, Shastri BJ, Prucnal PR. Broadcast and weight: An integrated network for scalable photonic spike processing. *J Lightwave Technol*. 2014;32(21):3427–3439.
59. Mourgias-Alexandris G, Totovic A, Tsakyridis A, Passalis N, Vyrsokinos K, Tefas A, Pleros N. Neuromorphic photonics with coherent linear neurons using dual-IQ modulation cells. *J Lightwave Technol*. 2019;38(4):811–819.
60. Zhu HH, Zou J, Zhang H, Shi YZ, Luo SB, Wang N, Cai H, Wan LX, Wang B, Jiang XD, et al. Space-efficient optical computing with an integrated chip diffractive neural network. *Nat Commun*. 2022;13(1):1044.
61. Al-Qadasi MA, Chrostowski L, Shastri BJ, Shekhar S. Scaling up silicon photonic-based accelerators: Challenges and opportunities. *APL Photonics*. 2022;7(2):020902.
62. Yann LeCun Y, Cortes C, Burges CJC. The MNIST database of handwritten digit. [accessed 2023 Jan 10] http://yann.lecun.com/exdb/mnist/
63. Ríos C, Youngblood N, Cheng Z, le Gallo M, Pernice WHP, Wright CD, Sebastian A, Bhaskaran H. In-memory computing on a photonic platform. *Sci Adv*. 2019;5(2):eaau5759.
64. Jouppi NP, Young C, Patil N, Patterson D, Agrawal G, Bajwa R, Bates S, Bhatia S, Boden N, Borchers A, et al. In-datacenter performance analysis of a tensor processing unit. In: *Proceedings of the 44th annual international symposium on computer architecture*. ISCA; 2017, p. 1–12.
65. Tait AN, de Lima TF, Zhou E, Wu AX, Nahmias MA, Shastri BJ, Prucnal PR. Neuromorphic photonic networks using silicon photonic weight banks. *Sci Rep*. 2017;7(1):1–10.
66. Bangari V, Marquez BA, Miller H, Tait AN, Nahmias MA, de Lima TF, Peng HT, Prucnal PR, Shastri BJ. Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs). *IEEE J Sel Top Quantum Electron*. 2019;26(1):1–13.
67. Xu X, Tan M, Corcoran B, Wu J, Boes A, Nguyen TG, Chu ST, Little BE, Hicks DG, Morandotti R, et al. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature*. 2021;589(7840):44–51.
68. Xu X, Tan M, Corcoran B, Wu J, Nguyen TG, Boes A, Chu ST, Little BE, Morandotti R, Mitchell A, et al. Photonic perceptron based on a Kerr microcomb for high-speed, scalable, optical neural networks. *Laser Photonics Rev*. 2020;14(10):2000070.
69. Bagherian H, Skirlo S, Shen Y, Meng H, Ceperic V, Soljacic M. On-chip optical convolutional neural networks. ArXiv. 2018. https://doi.org/10.48550/arXiv.1808.03303
70. Wetzstein G, Ozcan A, Gigan S, Fan S, Englund D, Soljačić M, Denz C, Miller DAB, Psaltis D. Inference in artificial intelligence with deep optics and photonics. *Nature*. 2020;588(7836):39–47.
71. Miscuglio M, Hu Z, Li S, George JK, Capanna R, Dalir H, Bardet PM, Gupta P, Sorger VJ. Massively parallel amplitude-only Fourier neural network. *Optica*. 2020;7(12):1812–1819.
72. Chang J, Sitzmann V, Dun X, Heidrich W, Wetzstein G. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Sci Rep*. 2018;8(1):12324.

73. Chen S, Wang X, Chen C, Lu Y, Zhang X, Wen L. DeepSquare: Boosting the learning power of deep convolutional neural networks with elementwise square operators. ArXiv. 2019. https://doi.org/10.48550/arXiv.1906.04979
74. Buchali F, Aref V, Dischler R, Chagnon M, Schuh K, Hettrich H, Bielik A, Altenhain L, Guntermann M, Schmid R, et al. 128 GSa/s SiGe DAC implementation enabling 1.52 Tb/s single carrier transmission. *J Lightwave Technol*. 2021;39(3):763–770.
75. Burla M, Hoessbacher C, Heni W, Haffner C, Fedoryshyn Y, Werner D, Watanabe T, Massler H, Elder D, Dalton L, et al. 500 GHz Plasmonic Mach-Zehnder Modulator. Paper presented at: 2019 Conference on Lasers and Electro-optics (CLEO); 2019 May; San Jose, CA.
76. Inference Platforms for HPC Data Centers from NVIDIA Deep Learning AI, NVIDIA. [accessed 2023 Jan 10] https://www.nvidia.com/en-au/deep-learning-ai/inference-platform/hpc/
77. Cloud TPU. System Architecture, Google Cloud. [accessed 2023 Mar 28] https://cloud.google.com/tpu/docs/system-architecture-tpu-vm
78. Hurtado A, Schires K, Henning ID, Adams MJ. Investigation of vertical cavity surface emitting laser dynamics for neuromorphic photonic systems. *Appl Phys Lett*. 2012;100(10):103703.
79. Robertson J, Wade E, Kopp Y, Bueno J, Hurtado A. Toward neuromorphic photonic networks of ultrafast spiking laser neurons. *IEEE J Sel Top Quantum Electron*. 2019;26(1):7700715.
80. Robertson J, Kirkland P, Alanis JA, Hejda M, Bueno J, di Caterina G, Hurtado A. Ultrafast neuromorphic photonic image processing with a VCSEL neuron. *Sci Rep*. 2022;12:4874.
81. Zhang Y, Robertson J, Xiang S, Hejda M, Bueno J, Hurtado A. All-optical neuromorphic binary convolution with a spiking VCSEL neuron for image gradient magnitudes. *Photonics Res*. 2021;9(5):B201–B209.
82. Rastegari M, Ordonez V, Redmon J, Farhadi A, XNOR-Net: Imagenet classification using binary convolutional neural networks. In: Leibe B, Matas J, Sebe N, Welling M, editors. *European conference on computer vision.* Amsterdam (Netherlands): Springer; 2016. p. 525–542.
83. Nahmias MA, Peng H-T, de Lima TF, Huang C, Tait AN, Shastri BJ, Prucnal PR. A laser spiking neuron in a photonic integrated circuit. ArXiv. 2020. https://doi.org/10.48550/arXiv.2012.08516
84. Barbay S, Kuszelewicz R, Yacomotti AM. Excitability in a semiconductor laser with saturable absorber. *Opt Lett*. 2011;36(23):4476–4478.
85. Selmi F, Braive R, Beaudoin G, Sagnes I, Kuszelewicz R, Barbay S. Relative refractory period in an excitable semiconductor laser. *Phys Rev Lett*. 2014;112(18):183902.
86. Mesaritakis C, Kapsalis A, Bogris A, Syvridis D. Artificial neuron based on integrated semiconductor quantum dot mode-locked lasers. *Sci Rep*. 2016;6:39317.
87. Sarantoglou G, Skontranis M, Bogris A, Mesaritakis C. Experimental study of neuromorphic node based on a multiwaveband emitting two-section quantum dot laser. *Photonics Res*. 2021;9(4):B87–B95.
88. Goulding D, Hegarty SP, Rasskazov O, Melnik S, Hartnett M, Greene G, McInerney JG, Rachinskii D, Huyet G. Excitability in a quantum dot semiconductor laser with optical injection. *Phys Rev Lett*. 2007;98(15):153903.
89. Alexander K, Van Vaerenbergh T, Fiers M, Mechet P, Dambre J, Bienstman P. Excitability in optically injected microdisk lasers with phase controlled excitatory and inhibitory response. *Opt Express*. 2013;21(22):26182–26191.
90. Yacomotti AM, Monnier P, Raineri F, Bakir BB, Seassal C, Raj R, Levenson JA. Fast thermo-optical excitability in a two-dimensional photonic crystal. *Phys Rev Lett*. 2006;97(14):143904.
91. Feldmann J, Youngblood N, Wright CD, Bhaskaran H, Pernice WHP. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature*. 2019;569:208–214.
92. Jha A, Huang C, Peng H-T, Shastri B, Prucnal PR. Photonic spiking neural networks and graphene-on-silicon spiking neurons. *J Lightwave Technol*. 2022;40(9):2901–2914.
93. Filipovich MJ, Guo Z, Al-Qadasi M, Marquez BA, Morison HD, Sorger VJ, Prucnal PR, Shekhar S, Shastri BJ. Silicon photonic architecture for training deep neural networks with direct feedback alignment. ArXiv. 2022. https://doi.org/10.48550/arXiv.2111.06862
94. Ren Q, Zhang Y, Wang R, Zhao J. Optical spike-timing-dependent plasticity with weight-dependent learning window and reward modulation. *Opt Express*. 2015;23(19):25247–25258.
95. Xiang S, Ren Z, Song Z, Zhang Y, Guo X, Han G, Hao Y. Computing primitive of fully VCSEL-based all-optical spiking neural network for supervised learning and pattern classification. *IEEE Trans Neural Netw Learn Syst*. 2021;32(6):2494–2505.
96. Mesaritakis C, Skontranis M, Sarantoglou G, Bogris A. Micro-ring-resonator based passive photonic spike-time-dependent-plasticity scheme for unsupervised learning in optical neural networks. Paper presented at: IEEE: Proceedings of the 2020 Optical Fiber Communications Conference and Exhibition (OFC); 2020 Mar 8–12; San Diego, CA.
97. Nahmias MA, Shastri BJ, Tait AN, Prucnal PR. A leaky integrate-and-fire laser neuron for ultrafast cognitive computing. *IEEE J Sel Top Quantum Electron*. 19(5):1800212.
98. Davies M, Wild A, Orchard G, Sandamirskaya Y, Guerra GAF, Joshi P, Plank P, Risbud SR. Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proc IEEE*. 2021;109(5):911–934.