RESEARCH-ARTICLE

Event-Driven Spatiotemporal Processing-In-Sensor with Phase Change Memory-based Optical Acceleration

MEHRDAD MORSALI, New Jersey Institute of Technology, Newark, NJ, United States

DENIZ NAJAFI, New Jersey Institute of Technology, Newark, NJ, United States

AMIN SHAFIEE, Colorado State University, Fort Collins, CO, United States

SEPEHR TABRIZCHI, University of Illinois at Chicago, Chicago, IL, United States

PIETRO MERCATI, Intel Corporation, Santa Clara, CA, United States

MOHSEN IMANI, University of California, Irvine, Irvine, CA, United States

View all

Open Access Support provided by:

New Jersey Institute of Technology

University of Illinois at Chicago

Intel Corporation

Colorado State University

University of California, Irvine

Advanced Micro Devices, Inc.

Published: 30 June 2025

Citation in BibTeX format

GLSVLSI '25: Great Lakes Symposium on VLSI 2025
June 30 - July 2, 2025
LA, New Orleans, USA

Conference Sponsors:
SIGDA

Event-Driven Spatiotemporal Processing-In-Sensor with Phase Change Memory-based Optical Acceleration

Mehrdad Morsali
New Jersey Institute of Technology
Newark, USA
mm2772@njit.edu

Deniz Najafi
New Jersey Institute of Technology
Newark, USA
dn339@njit.edu

Amin Shafiee
Colorado State University
Fort Collins, USA
amin.shafiee@colostate.edu

Sepehr Tabrizchi
University of Illinois Chicago
Chicago, USA
ssohra2@uic.edu

Pietro Mercati
Intel Corporation.
Hillsboro, USA
pietromercati@gmail.com

Mohsen Imani
University of California-Irvine
Irvine, USA
m.imani@uci.edu

Arman Roohi
University of Illinois Chicago
Chicago, USA
aroohi@uic.edu

Navid Khoshavi
AMD
Orlando, USA
Navid.Khoshavi@amd.com

Mahdi Nikdast
Colorado State University
Fort Collins, USA
Mahdi.Nikdast@colostate.edu

Shaahin Angizi
New Jersey Institute of Technology
Newark, USA
shaahin.angizi@njit.edu

Abstract

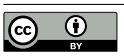
This work introduces a novel hybrid electronic-optical processing-in-sensor architecture designed for low-cost, real-time frame processing at the edge. The proposed system enables event detection and integrates a TinyLSTM-based temporal inference model to analyze multiple frames in real time, extracting meaningful spatiotemporal features that trigger an address actuator for region-of-interest selection. By selectively reading out only relevant pixel regions, the architecture significantly reduces data transfer overhead and power consumption. Additionally, it harnesses the efficiency of silicon photonic (SiPh) devices to enable adaptive frame compression techniques and perform convolution operations through intrinsic, conversion-free multiply-accumulate computations. Device-to-architecture simulation results demonstrate 11.2× improvement in performance compared to the state-of-the-art SiPh accelerator achieving 37 KFPS/W. This marks a significant advancement in processing-in-sensor technology, enhancing both computational efficiency and energy savings for edge AI applications.

CCS Concepts

• **Hardware** → *Emerging optical and photonic technologies.*

Keywords

Vision Sensors, Deep Neural Networks, Processing-In-Sensor, Silicon Photonics



GLSVLSI '25, New Orleans, LA, USA

ACM ISBN 979-8-4007-1496-2/25/06
https://doi.org/10.1145/3716368.3735243

ACM Reference Format:
Mehrdad Morsali, Deniz Najafi, Amin Shafiee, Sepehr Tabrizchi, Pietro Mercati, Mohsen Imani, Arman Roohi, Navid Khoshavi, Mahdi Nikdast, and Shaahin Angizi. 2025. Event-Driven Spatiotemporal Processing-In-Sensor with Phase Change Memory-based Optical Acceleration. In *Great Lakes Symposium on VLSI 2025 (GLSVLSI '25), June 30–July 02, 2025, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3716368.3735243

1 Introduction

The rapid expansion of the Internet of Things (IoT) has enabled real-time data processing, with vision sensors playing a crucial role in applications such as smart surveillance, autonomous navigation, etc. These sensors continuously capture vast amounts of visual data, which traditionally rely on cloud-based architectures for processing. However, such systems suffer from critical challenges, including high power consumption, increased latency, and scalability constraints. A major contributor to these inefficiencies in IoT vision systems is the traditional imaging pipeline, which consists of a CMOS image sensor (CIS) and a back-end image signal processor (ISP), as shown in Fig. 1(a). In such systems, off-chip communication remains a significant energy overhead, where Bluetooth Low Energy (BLE) requires 1 nJ/bit, whereas a Multiply-Accumulate (MAC) operation consumes less than 0.2 pJ/8-bit [4, 5].

Edge Intelligence has emerged as a promising solution by enabling local processing at IoT nodes, reducing data transmission, and improving efficiency. Enabled by Processing-in-sensor (PIS) architecture, a more efficient alternative by integrated computation directly within the sensor, thereby minimizing unnecessary data movement. Unlike conventional architectures, PIS extracts relevant features at the sensor level, transmitting only essential information while discarding redundant raw data. This method significantly

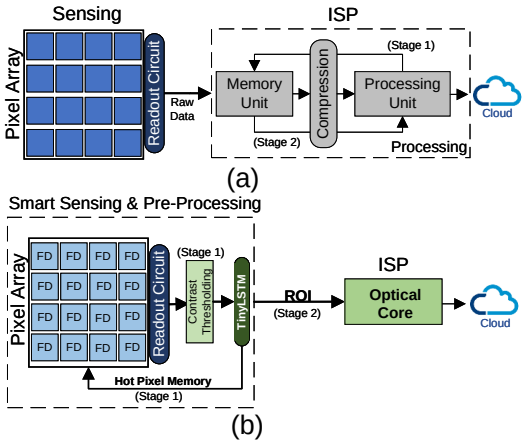


Figure 1: (a) Conventional cloud-based ROI detector vs. (b) Proposed approach that minimizes unnecessary data transfer between the imager and the ISP.

reduces energy consumption, enhances real-time responsiveness, and alleviates memory bandwidth constraints.

Although PIS is activated and low-power sensors are used, continuous image recording and streaming place a significant demand on bandwidth, quickly consuming energy resources [27]. To mitigate this, object detection is widely employed to analyze data locally before initiating recording and transmission, thereby improving power efficiency [6]. However, continuous object detection itself remains power-intensive. Consequently, event-driven vision sensors have gained traction as triggers, effectively minimizing redundant detections while ensuring target objects are not missed [9]. In [10], an intelligent vision sensor is presented, integrating a tiny CNN model and a programmable PIS circuit to enable real-time inference for low-power edge devices. In [8], a multimode vision sensor is presented, integrating a PIS technique to facilitate the efficient extraction of both temporal and spatial information. The proposed architecture employs an innovative temporal contrast pixel to capture the contrast between successive frames. Furthermore, to alleviate the high power consumption associated with analog-to-digital converter (ADC) operations, the design utilizes local binary pattern extraction. Additionally, to further reduce power consumption, [15] proposes a resource-efficient neural-network-based face detection system. This system utilizes 1.5-bit frame-to-frame delta quantization combined with a diagonal spatial feature extraction method, specifically designed for resource-constrained. To the best of our knowledge, most vision sensor designs based on PIS implementation primarily utilize electronic components for tasks such as neural network implementation. However, Silicon Photonics (SiPh) based designs offer significantly higher operational bandwidths and more favorable resolution fan-in/fan-out characteristics, making them a preferred choice for neural network implementation [13].

In this paper, we introduce a novel intelligent processing-in-sensor architecture as depicted in Fig. 1(b) designed for resource-constrained, always-on event-driven camera sensors. Our approach enables low-power analog-domain frame differencing for motion sensing while integrating a TinyLSTM-based temporal inference model that analyzes multiple frames to extract meaningful spatiotemporal features. These extracted features dynamically trigger an address actuator, selectively reading out only the region of interest (ROI) from the pixel array. Furthermore, an adaptive frame compression mechanism or a convolution layer implementation using SiPh core is incorporated to facilitate efficient on-chip or off-chip feature extraction and object detection. The key contributions of this work are as follows:

(1) We develop a specialized processing framework that leverages a set of optimized circuit-level techniques, including an upgraded pixel structure, low-power analog thresholding, and TinyLSTM-based temporal modeling to enable high-speed analog-domain event detection. By utilizing a lightweight LSTM model, the system effectively learns temporal dependencies to distinguish meaningful motion from transient noise, improving system responsiveness while minimizing redundant data processing. (2) We design a SiPh processing core capable of supporting adaptive image compression strategies and event-driven address actuation. This enables efficient analog-domain preprocessing of selected ROI frames, significantly reducing data bandwidth while optimizing downstream processing efficiency. The combination of TinyLSTM-driven event selection and SiPh-based optical acceleration ensures an energy-efficient pipeline for real-time inference. (3) We construct a comprehensive evaluation framework, bridging circuit-level characteristics to system-level architecture, to rigorously assess and optimize our proposed design.

2 Background

Event-Driven Vision Sensors. In image sensor systems, transferring large volumes of bulky data between the sensor and processor increases latency and power consumption. However, motion detection enables intelligent power management by transmitting only relevant features instead of the entire raw dataset, enhancing efficiency. For in-pixel processing, two primary approaches are employed to extract temporal features: the Dynamic Vision Sensor (DVS) and the frame-based Frame Difference (FD) vision sensor. The DVS approach continuously monitors photocurrent contrast without performing signal integration. To detect subtle temporal variations, an additional gain stage in the sensing frontend is required, leading to increased power consumption and a reduced fill factor (FF). Additionally, DVS provides motion-triggered pixel addresses rather than full exposure data, making post-processing more complex and less adaptable to diverse applications.

In contrast, the FD-based approach captures changes in exposure between consecutive frames. By leveraging inherent signal amplification through integration instead of an external gain stage, the FD sensor achieves higher sensitivity and operates with lower power consumption. Moreover, its output is more compatible with mainstream signal processing architectures, including neural networks[8].

Silicon Photonics Acceleration. In deep neural network (DNN) implementations, SiPh-based accelerators [3, 12–14] offer promising advantages due to their higher operational bandwidth and enhanced resolution. These accelerators are classified into coherent and non-coherent implementations, each offering distinct advantages. In the coherent implementations, a single wavelength carries all weight and activation-modulated signals, which are encoded into the phase, the electrical field amplitude, or the polarization of the optical signal. In contrast, non-coherent designs utilize multiple

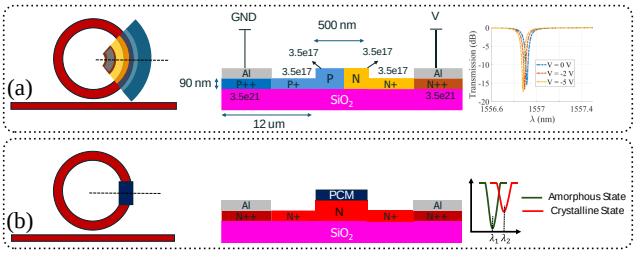


Figure 2: (a) PN-doped junction in reverse bias integrated with MR for signal modulation. (b) MR integrated with PCM for nonvolatile optical modulation.

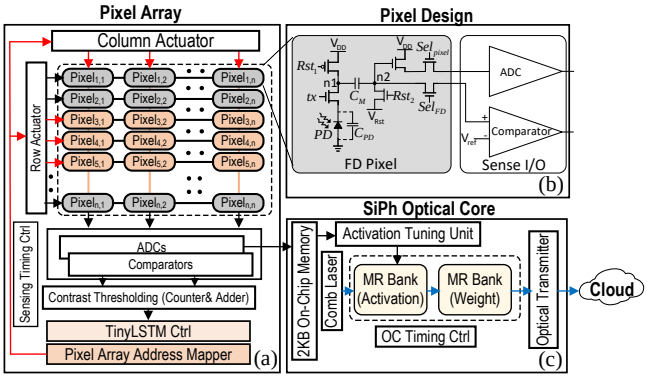


Figure 3: Block-level implementation of the proposed event-driven PIS architecture. (a) Pixel Array and its peripherals (b) FD pixel and reading circuitry, (c) Optical processing core.

wavelengths to enable parallel computations, encoding parameters within the signal's amplitude. In these designs, precise tuning of Micro-ring Resonators (MRs) is essential for managing specific wavelengths. Acting as weights, MRs modulate the intensity of incoming light at designated wavelengths. The implementation of the MAC operation, essential for neural network processing, is achieved by tuning the resonant wavelengths of MRs to overlap with the input light's wavelength, thereby embedding the desired parameters into the transmission spectrum. Many studies focus on designing DNNs that leverage SiPh-based accelerators. However, most suffer from the high-power consumption of ADC and DAC units, which also contribute to increased area occupation. Moreover, some studies explore replacing ADCs with MR-based adders and shifters. However, the extensive use of MRs for encoding both activation and weight parameters increases delays and power consumption, reducing the system's adaptability for various DNN applications [12, 14]. In our design, to enhance power efficiency, the MRs allocated for weight value modulation are equipped with PCM, making them non-volatile and reducing tuning energy, as the weights of a neural network are mostly constant.

MR Devices. The resonance peak of SiPh MRs can be shifted left or right by inducing slight perturbations in the accumulated round-trip phase within the ring region. This can be achieved using the thermo-optic or electro-optic effect or by integrating the MR with an optical material such as phase-change material (PCM). In our design, we use two types of MRs. To enable fast application of activation parameters, we integrated a reverse-biased PN-phase shifter into the MRs. The designed active ring is shown in Fig. 2(a). It has a radius of 18.48 μm and a modulation fraction of 80%. Lumerical simulations indicate an electrical bandwidth of up to 40 GHz, a Free Spectral Range (FSR) of 5 nm, an extinction ratio of ~ 18 dB, and a Q-factor of ~300. Additionally, simulations show an on-resonance insertion loss of about 0.51 dB due to the free-carrier absorption (FCA) effect.

As for weight parameters, we opted to integrate the passive rings with PCMs (depicted in Fig. 2(b)) to leverage their nonvolatile properties. PCMs are a class of optical materials that can repeatedly change their phase from amorphous (crystalline) to crystalline (amorphous) states in a nonvolatile manner and their optical and electrical properties change as a function of their phase state [18, 19]. Despite their ability to change phase states in a nonvolatile manner, PCMs require slow and power-intensive mechanisms to reshape their molecular structure using an external heat source. Therefore, in this work, we integrated the Sb_2Se_3 —a well-known PCM with a low loss of up to 0.018 $dB/\mu m$— into the MRs responsible for imprinting the weight parameters, as these parameters do not require frequent updates. As the phase state of the PCM-on-MR changes through the application of a voltage signal to the underlying doped heater, the MR's resonance peak shifts left or right, enabling precise tuning according to the weight parameters. [16]. Note that the PCM-loaded MRs only need a one-time calibration and due to the nonvolatile optical properties of the PCMs, the static power consumption of the MR will be zero.

3 Proposed Architecture

3.1 Microarchitecture Design

Pixel array. The presented PIS architecture primarily comprises a pixel array and an Optical core, as illustrated in Fig. 3. The pixel array shown in Fig. 3(a), consists of 126 × 126 FD pixels along with peripherals including readout circuitry, contrast thresholding, and tiny LSTM. It is responsible for capturing images while consistently extracting temporal information by comparing exposure values from consecutive frames. The FD pixel, shown in Fig. 3(b), is a simplified version of the pixel presented in [8], which was originally designed for a multimode vision sensor. In our case, we are only utilizing it for FD purposes with a comparator-based readout circuit. A memory capacitor (C_m) has been added to the structure of a conventional pixel and by following a specific operation flow, the FD pixel generates a voltage corresponding to the voltage difference of the photodiode between two consecutive exposures. C_m is responsible for storing frame information and performing the subtraction of consecutive frames using the inherent characteristics of the capacitor. First, C_m is connected to the photodiode (PD) at node n1, while node n2 is connected to V_{Rst}, allowing charge sharing to occur. Thus, n2 will have a fraction of the voltage drop from the first frame (F1). Then, n2 will be float while the PD is reset. After that, the second frame (F2) exposure takes place while n2 remains floating. Consequently, n1 experiences a voltage drop of F2, and the floating node n2 will hold a ratio of F1 - F2. A detailed explanation of the FD pixel can be found in reference [8]. For FD readout, the FD pixels in a column are connected to a comparator with a specific reference voltage. If the measured value exceeds the threshold—indicating a difference between consecutive frames—the

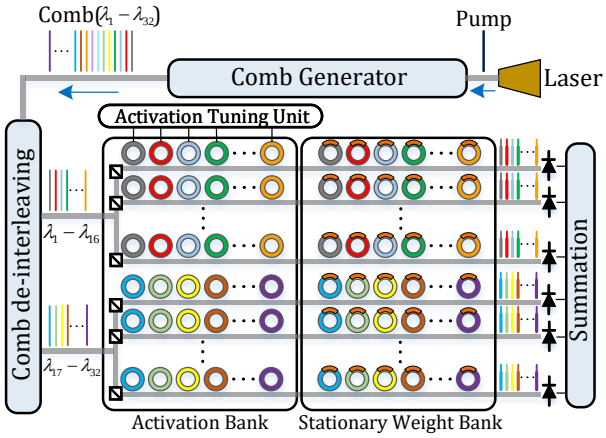


Figure 4: SiPh optical processing core.

comparator output will be '1'. The outputs of the comparators are connected to 6-bit counters in the contrast thresholding circuit, and an adder sums the values of all counters to determine the total number of pixels that have changed. If this number exceeds a predefined threshold—determined based on the ROI size and application—it is considered an event.

SiPh Optical Core. The architecture of the optical core is illustrated in detail in Fig. 4. The optical core consists of a comb laser generator, which serves as the source of light signals, a de-interleaver that divides the comb into smaller groups, two MR banks for activation and weight modulation, and a summation section for combining intermediate results when needed. The comb laser considered here was introduced in [17] and is capable of generating 32 independent wavelength channels. A one-stage de-interleaver splits these 32 wavelengths into two groups of 16, as each arm in our MR banks contains 16 MRs with independent operational wavelengths. Further subdivision of these 16-wavelength groups is performed using splitters, allowing signals to be sent to multiple arms of the banks, thereby increasing the computational capacity of the optical processing core. The core's main computational components are two MR banks: a PN-junction-based MR bank and a PCM-MR bank. As mentioned earlier, the former is used for activation modulation, while the latter handles weight modulation. Since the network's weights remain fixed over time, they are mapped onto the PCM-MRs that are non-volatile at the start of the core's operation. Once set, only the activation values need to be modulated on the MRs in the activation bank. The optical core performs the MAC operation by altering the intensity of light signals according to activation and weight values using MRs, in a parallel manner across multiple wavelengths. Balanced Photodetectors carry out the accumulation of all multiplication values at the end of each arm. This computational capability can be utilized for implementing NN layers, compression layers, or any other application that relies on MAC operations.

TinyLSTM Ctrl. TinyLSTM Controller functionality is inspired by [20],[7], and [26] for object trajectory prediction. It enables the model to capture motion patterns in multiple frames. TinyLSTM processes spatiotemporal data by extending traditional convolutional layers and incorporating LSTM units. The ConvLSTM with latent texture encoding efficiently predicts object trajectory by first transforming raw pixel data into a lower-dimensional feature space before applying ConvLSTM for motion tracking. Our network topology in Section 4 encodes each 126×126 grayscale frame into a compact 4×4×16 latent texture using a series of depthwise-separable convolutional layers with aggressive downsampling. For the LSTM acceleration, we adopted the SRAM accelerator in [21].

3.2 Software Support

Core Algorithm. We propose an algorithm to optimize frame processing in event-driven vision sensors by dynamically selecting among three operational modes: *Event_Detection*, *ROI_ Prediction*, and *Optic_Processing*. Before detailing the procedures described in Algorithm 1, we first explain two key functions. Our design, based on the pixel array structure and its peripherals, incorporates two types of read functions: High-Resolution Read (`Highres_read`) and Frame Difference Read (`FD_Read`). `Highres_Read` involves reading the actual pixel values using ADCs and transmitting them to the optical core or TinyLSTM. In contrast, `FD_Read` detects and analyzes frame differences using comparators and thresholding circuitry for event detection.

Algorithm 1 Presented Event-Detection Algorithm

```
Input1: Time_out (T_τ)
Output1: Operational_mode
Output2: ROI_address
procedure EVENT-DETECTION
    if ROI_Flag = '0' then
        Read_address ← Boundary, Event ← FD_Read ()
        if Event = '1' then enable ROI_PREDICTION
        end if
    else
        Read_address ← ROI_address, Timer ()
        if time < T_τ then
            Event ← FD_Read ()
            if Event = '1' then enable OPTIC_PROCESSING
            end if
        else
            Timeout ← '1', enable ROI_PREDICTION
        end if
    end if
end procedure
procedure ROI_PREDICTION
    Read_address ← Full Frame, F ← Highres_Read ()
    Coordinate, New_ROI ← LSTM ()
    if Timeout = '0' ∨ New_ROI = '1' then ROI_Address ← Coordinate, ROI_Flag ← '1'
    elseROI_Flag ← '0', enable EVENT_DETECTION
    end if
end procedure
procedure OPTIC_PROCESSING
    Read_address ← ROI_address, F_out ← Highres_Read () enable ROI_PREDICTION
end procedure
```

Fig. 5 depicts three primary procedures—*Event_Detection*, *ROI_ Prediction*, and *Optic_Processing*—work together to dynamically adjust the sensor's operational mode based on detected events. At the start of runtime, the *Event_Detection* procedure performs `FD_Read` on only the boundary areas of the frame to detect any entering events. This significantly reduces energy consumption by activating only the necessary parts of the frame. If an event is detected, the system switches to *ROI_Prediction* mode, where TinyLSTM is activated. During this mode, each frame's pixel values are stored in the frame buffer, with the memory address of each pixel within the

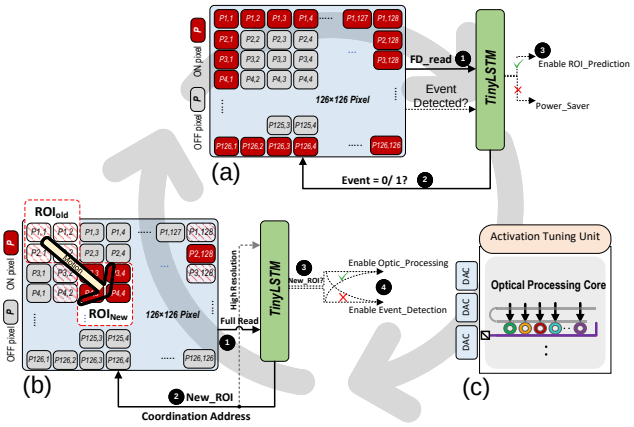


Figure 5: (a) Event detection mode, (b) ROI prediction mode, and (c) Optic processing mode.

frame calculated based on its coordinates in the pixel array. Three consecutive frames are read using *Highres_Read*, and instead of reloading the entire frame buffer, only the memory content for the pixels within the ROI is updated. TinyLSTM then analyzes these frames to predict the object's trajectory and generates coordinates for an area where the object is most likely to enter. The ROI is defined as the current area where the object is located plus the area where it is most likely to enter.

After extracting the ROI, the system switches back to *Event_Detection* mode, this time performing `FD_Read` within the predicted ROI. A timer is started, and the ROI is monitored for a specific period to check for an event. If no event is detected—indicating either an incorrect ROI prediction or a stationary object—the system reverts to *ROI_Prediction* mode to determine a new, more accurate ROI. If a new ROI is identified, the algorithm updates the ROI address and switches to *Event_Detection* within the newly defined area, otherwise, like the beginning of processing, *Event_Detection* will be run on boundary areas. Otherwise, if an event is detected before timeout, the system switches to *Optic_Processing* mode. Here, `Highres_read` is performed on the ROI, and the frame data is sent to the optical core for further processing. Immediately after, the system returns to *ROI_Prediction* mode, where TinyLSTM tracks changes and predicts a new ROI.

Frame Parallelism. To optimize resource utilization and improve processing efficiency, we introduce frame parallelism in the proposed event-driven vision sensor. This is achieved through a fully pipelined in-sensor computation optimization, which leverages temporal parallelism to ensure that multiple frames are processed concurrently at different pipeline stages. If fully processed, each frame undergoes thirteen equal processing operations within the pipeline as shown in Fig. 6, ensuring balanced execution across all hardware components. The slowest processing stage dictates the execution time per step, ensuring uniform workload distribution and preventing bottlenecks.

The effective throughput is determined by the mean frame execution time, $\mathbb{E}[T_f]$, with the throughput defined as $\frac{1}{\mathbb{E}[T_f]}$. In the assumed workload, approximately 50% of the frames trigger an early exit, thus bypassing part of the pipeline, while the remainder undergoes near-full processing. Notably, early-exiting frames may

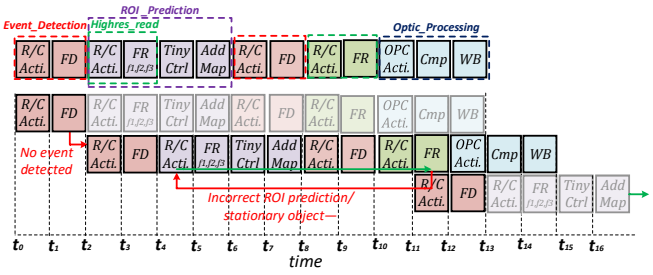


Figure 6: Frame parallelism idea in the event-driven sensor.

be completed in either 2 or 12 cycles. Assuming these two early-exit durations occur with equal probability, the expected execution time for an early-exiting frame is: $\mathbb{E}[T_{\text{early}}] = 0.5 \times 2 + 0.5 \times 12 = 7$ cycles. Fully processed frames, on the other hand, require 13 cycles. Consequently, the overall expected frame execution time becomes $\mathbb{E}[T_f] = 0.5 \times 7 + 0.5 \times 13 = 10$ cycles. This multi-frame parallel processing strategy, which leverages speculative execution and dynamic scheduling to trigger early terminations when feasible, significantly enhances pipeline utilization and reduces overall latency. As a result, even in power-constrained environments, the vision system achieves high efficiency, making real-time, event-driven visual sensing both technically and operationally viable.

4 Experimental Results

Bottom-up Evaluation Framework. The proposed evaluation framework, depicted in Fig. 7, spans multiple design hierarchies, covering device, device, circuit, architecture, and application levels. At the device level, we designed and simulated PCM-MR devices in Lumerical MODE solver [1], extracting device parameters for seamless co-simulation with interface CMOS circuits in SPICE. Progressing to the circuit level, we first designed the pixel array and peripheral circuitry using the 45nm NCSU Product Development Kit (PDK) library [2] in HSPICE, enabling the extraction of output voltage and current characteristics. We then implemented optical core components in HSPICE, except for the on-chip memory, which was modeled using Cacti [25]. At the application level, the weight parameters for object detection are extracted, quantized, and scaled before being mapped onto the PCM-MR devices. At the architectural level, we developed a dedicated simulator tailored to our platform, facilitating the evaluation of the execution time and energy efficiency of our platform.

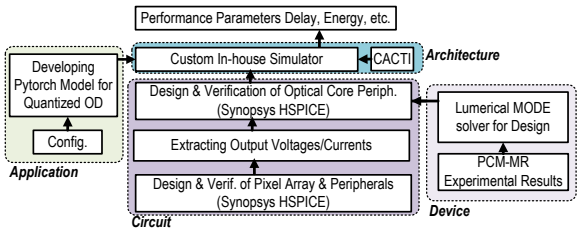


Figure 7: Proposed bottom-up evaluation framework.

TinyLSTM for Capturing Motion Patterns. Here we provide a detailed architectural description of the TinyLSTM model employed in this work, including its network topology and layer configurations. Each frame from the pixel array is processed through five

DS-Conv blocks as shown in Table 1, resulting in ~1.05 million MACs per frame. The sequence of latent textures—each of size 4×4×16—is then fed into two ConvLSTM layers as presented in Table 1. They use 3×3 kernels and output 16 channels, with each layer requiring about 442,000 MACs (accounting for the four gating operations in LSTM cells). The final state from the ConvLSTM, having dimensions 4×4×16, is flattened to a 256-dimensional vector and processed by a dense layer that reduces it to 128 units (~262K MACs), followed by an output dense layer that maps to 2 units representing the predicted (x, y) coordinates of the object's next position. In total, the complete pipeline requires roughly 4.3 million MAC operations, making it efficient for real-time trajectory prediction on resource-constrained IoT devices. To further reduce energy consumption, INT8 quantization is used instead of FP32, achieving a 4× reduction in energy per MAC operation.

Table 1: TinyLSTM Model Layers.

Layer	Input	Operation	Filters	Output
Depthwise Separable Convolutional Layers				
DS-Conv1	126×126×1	3×3 Depthwise Conv	16	63×63×16
DS-Conv2	63×63×16	3×3 Depthwise Conv	16	32×32×16
DS-Conv3	32×32×16	3×3 Depthwise Conv	16	16×16×16
DS-Conv4	16×16×16	3×3 Depthwise Conv	16	8×8×16
DS-Conv5	8×8×16	3×3 Depthwise Conv	16	4×4×16
ConvLSTM Layers				
ConvLSTM1	(T=3, 4×4×16)	3×3 ConvLSTM (gating factor 4)	16	4×4×16
ConvLSTM2	(T=3, 4×4×16)	3×3 ConvLSTM (gating factor 4)	16	4×4×16
Fully Connected (Dense) Layers				
Dense1	Flattened 256	Fully Connected	128	128
Dense2 (Output)	128	Fully Connected	2	2

Power and Energy. To demonstrate the efficiency of the proposed architecture and event-driven algorithm, we conducted two sets of power and energy consumption analyses. As mentioned earlier, our design operates in three phases: event detection, ROI prediction, and optic processing. The maximum power consumption during these phases is shown in Fig. 8(a). The figure presents two power values for event detection: one for detecting events on the frame boundary and another for a 64×64 ROI. Overall, event detection has the lowest power consumption, especially at the boundary. According to Fig. 8(a), optic processing consumes significantly more power than event detection and ROI prediction, primarily due to the energy required for tuning MRs to modulate activation values. Therefore, reducing time spent in optic processing mode leveraging event detection mechanism, conserves power. Also, in ROI prediction mode, the power consumption of pixels is increased compared to the other two, as we need to read the entire frame.

Fig. 8(b) demonstrates the efficiency of processing the ROI instead of the full frame by presenting the optical core's energy consumption while performing a convolution layer with a 3×3 kernel and a 4×4 average pooling compression layer on frames of varying sizes. The plots indicate the percentage of energy reduction achieved when processing smaller frames instead of the full frame.

Performance. The performance of the design in different modes has been provided here. In event detection mode, the frame/s performance varies depending on the size of the monitored ROI. An average performance of 2.6×10^6 frame/s can be achieved, considering the average operation time for ROI sizes of 16×16, 32×32, and 64×64. In ROI prediction mode, the TinyLSTM processor predicts

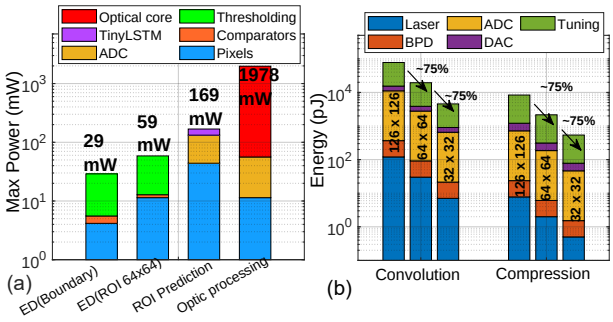


Figure 8: (a) Power consumption by operational phase. (b) Optical core energy consumption for convolution and compression layers across varying frame sizes.

the ROI by processing three consecutive frames in 9.18 μs. Finally, the optical processing core achieves a performance of 64 GOPS/W.

Comparison vs. SiPh Edge Accelerators. In Table 2 we compare the efficiency of various MR-based optical accelerators, including LightBulb [28], HolyLight [11], HQNNA [23], Robin [24], and CrossLight [22], for edge integration with the presented design. Results indicate that the proposed hybrid design, which incorporates an electronic front-end for event detection and ROI prediction, significantly outperforms comparable designs. Specifically, it enhances performance by 243.2%, 6.94%, 1021.2% over CrossLight [22], HQNNA [23], and HolyLight [11], respectively, despite considering the pre-processing of each frame through both event detection and ROI prediction steps. While Light-Bulb [28] and Robin [24] demonstrate higher performance, they lack event detection and ROI prediction capabilities.

Table 2: Comparison with SOTA optical accelerators.

Designs	LightBulb [28]	HolyLight [11]	HQNNA [23]	Robin [24]	CrossLight [22]	Proposed
Node (nm)	32	32	45	45	*	45
KFPS/W	57.75	3.3	34.6	46.5	10.78	37
Improv.	-35.9%(↓)	1021.2%(↑)	6.94%(↑)	-20.4%(↓)	243.2%(↑)	ref
OC	✓	✓	✓	✓	✓	✓
ED	✗	✗	✗	✗	✗	✓
ROI	✗	✗	✗	✗	✗	✓

OC: Object Classification, ED: Event Detection, ROI: Region of Interest Prediction. *Data is not reported/not achievable from the paper [22].

5 Conclusion

This work introduces an event-driven processing-in-sensor architecture that combines TinyLSTM-based temporal inference with SiPh acceleration for efficient, real-time frame preprocessing at the edge. Using a lightweight LSTM model, it extracts spatiotemporal features and selectively reads out only the ROI, reducing data bandwidth and power consumption. The SiPh processing core enables adaptive frame compression and MAC operations for fast, energy-efficient vision processing in resource-constrained environments. While in this work we mainly focused on the hardware realization of the proposed algorithm, Future research will refine the TinyLSTM model for improved event detection with minimal computational overhead and explore adaptive optical computing to enhance scalability for diverse edge AI workloads. Simulation results show an 11.2× performance boost over the state-of-the-art SiPh accelerator, achieving 37 KFPS/W while incorporating event detection and ROI prediction features.

Acknowledgments

This work is supported in part by the National Science Foundation (NSF) under grant numbers 2216772, 2228028, 2401537, 2046226, 2448133, 2504839, 2447566, and Semiconductor Research Corporation (SRC).

References

[1] 2011. *Ansys Lumerical. [Online].* Available:https://www.lumerical.com/products/
[2] 2011. *NCSU EDA FreePDK45.* https://eda.ncsu.edu/freepdk/freepdk45/
[3] Salma Afifi et al. 2023. GHOST: A graph neural network accelerator using silicon photonics. *ACM TECS* 22 (2023).
[4] Ian F Akyildiz et al. 2002. Wireless sensor networks: a survey. *Computer networks* 38, 4 (2002), 393–422.
[5] Kenneth C Barr and Krste Asanović. 2006. Energy-aware lossless data compression. *ACM Transactions on Computer Systems (TOCS)* 24, 3 (2006), 250–291.
[6] Kyeongryeol Bong et al. 2018. A Low-Power Convolutional Neural Network Face Recognition Processor and a CIS Integrated With Always-on Face Detector. *IEEE JSSC* 53, 1 (2018), 115–123.
[7] Qinyu Chen et al. 2023. 3ET: Efficient Event-based Eye Tracking using a Change-Based ConvLSTM Network. In *BioCAS*. 1–5.
[8] Min-Yang Chiu et al. 2023. A Multimode Vision Sensor With Temporal Contrast Pixel and Column-Parallel Local Binary Pattern Extraction for Dynamic Depth Sensing Using Stereo Vision. *IEEE JSSC* 58 (2023).
[9] Kyojin D. Choo et al. 2019. 5.2 Energy-Efficient Low-Noise CMOS Image Sensor with Capacitor Array-Assisted Charge-Injection SAR ADC for Motion-Triggered Low-Power IoT Applications. In *ISSCC*.
[10] Tzu-Hsiang Hsu et al. 2023. A 0.8 V Intelligent Vision Sensor With Tiny Convolutional Neural Network and Programmable Weights Using Mixed-Mode Processing-in-Sensor Technique for Image Classification. *IEEE JSSC* 58 (2023).
[11] Weichen Liu et al. 2019. Holylight: A nanophotonic accelerator for deep learning in data centers. In *DATE*. IEEE, 1483–1488.
[12] Mehrdad Morsali et al. 2024. Lightator: An optical near-sensor accelerator with compressive acquisition enabling versatile image processing. In *DAC*. 1–6.
[13] Mehrdad Morsali et al. 2024. OISA: Architecting an Optical In-Sensor Accelerator for Efficient Visual Computing. In *DATE*. 1–6.
[14] Deniz Najafi et al. 2025. Neuro-Photonix: Enabling Near-Sensor Neuro-Symbolic AI Computing on Silicon Photonics Substrate. *IEEE TCASAI* (2025).
[15] Ning Pu et al. 2023. Resource-efficient Face Detector Using 1.5-bit Frame-to-frame Delta Quantization for Image Based Always-on Wake-up Application. In *ISCAS*.
[16] Carlos Ríos et al. 2022. Ultra-compact nonvolatile phase shifter based on electrically reprogrammable transparent phase change materials. *PhotoniX* 3 (2022).
[17] Anthony Rizzo et al. 2023. Massively scalable Kerr comb-driven silicon photonic link. *Nature Photonics* 17, 9 (2023), 781–790.
[18] Amin Shafiee et al. 2024. Programmable phase change materials and silicon photonics co-integration for photonic memory applications: a systematic study. *Journal of Optical Microsystems* 4 (2024).
[19] Amin Shafiee, Sudeep Pasricha, and Mahdi Nikdast. 2023. A survey on optical phase-change memory: The promise and challenges. *IEEE Access* 11 (2023), 11781–11803.
[20] Xiao Song et al. 2021. Pedestrian Trajectory Prediction Based on Deep Convolutional LSTM Network. *IEEE TITS* 22 (2021).
[21] Amitesh Sridharan et al. 2022. A 1.23-ghz 16-kb programmable and generic processing-in-sram accelerator in 65nm. In *ESSCIRC*. IEEE, 153–156.
[22] Febin Sunny et al. 2021. CrossLight: A cross-layer optimized silicon photonic neural network accelerator. In *DAC*. IEEE.
[23] Febin Sunny et al. 2022. A silicon photonic accelerator for convolutional neural networks with heterogeneous quantization. In *GLSVLSI*. 367–371.
[24] Febin P. Sunny et al. 2021. ROBIN: A robust optical binary neural network accelerator. *ACM TECS* 5s (2021).
[25] Shyamkumar Thoziyoor, Naveen Muralimanohar, Jung Ho Ahn, and Norman P Jouppi. 2008. *CACTI 5.1.* Technical Report. Technical Report HPL-2008-20, HP Labs.
[26] Tizian Zeltner et al. 2024. Real-time Neural Appearance Models. *ACM Trans. Graph.* 43 (2024).
[27] Xiaopeng Zhong et al. 2020. A Fully Dynamic Multi-Mode CMOS Vision Sensor With Mixed-Signal Cooperative Motion Sensing and Object Segmentation for Adaptive Edge Computing. *IEEE JSCC* 55 (2020).
[28] Farzaneh Zokaee et al. 2020. LightBulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks. In *DATE*. IEEE, 1438–1443.