TizBin: A Low-Power Image Sensor with Event and Object Detection Using Efficient Processing-in-Pixel Schemes

Sepehr Tabrizchi*, Shaahin Angizi†, Arman Roohi*
*School of Computing, University of Nebraska–Lincoln, Lincoln NE, USA
†Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA
aroohi@unl.edu, shaahin.angizi@njit.edu

2022 IEEE 40th International Conference on Computer Design (ICCD) | 978-1-6654-6186-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCD56317.2022.00117

***Abstract*—In the Artificial Intelligence of Things (AIoT) era, always-on intelligent and self-powered visual perception systems have gained considerable attention and are widely used. Thus, this paper proposes TizBin, a low-power processing in-sensor scheme with event and object detection capabilities to eliminate power costs of data conversion and transmission and enable data-intensive neural network tasks. Once the moving object is detected, TizBin architecture switches to the high-power object detection mode to capture the image. TizBin offers several unique features, such as analog convolutions enabling low-precision ternary weight neural networks (TWNN) to mitigate the overhead of analog buffer and analog-to-digital converters. Moreover, TizBin exploits non-volatile magnetic RAMs to store NN's weights, remarkably reducing static power consumption. Our circuit-to-application co-simulation results for TWNNs demonstrate minor accuracy degradation on various image datasets, while TizBin achieves a frame rate of 1000 and efficiency of ∼1.83 TOp/s/W.**

1. Introduction

Internet of Things (IoT) devices are expected to reach $1100B in revenue by 2025, with a web of interconnections estimated to consist of approximately 75+ billion IoT devices, including wearable devices as well as smart cities and industries [1], [2]. Artificial Intelligence of Things (AIoT) nodes are composed of a variety of sensors, which are used to collect and process data from the environment and people. There is usually a great deal of redundant and unstructured sensory data captured. The conversion and transmission of large raw data to a backend processor are energy-intensive, high-latency, a memory bottleneck, and low-speed feature extraction at the edge [1], [3]. Those issues can be addressed by shifting computing architecture from a cloud-centric way of thinking to a thing-centric (data-centric) perspective, where IoT nodes process sensed data. Despite such challenges, artificial intelligence tasks that require hundreds of layers of Convolutional Neural Networks (CNNs) have severe computational and storage constraints. There has been considerable advancement in both software and hardware to improve CNN efficiency by mitigating the "power and memory wall" bottleneck.

There has been considerable exploration of shallower but wider CNN models, quantizing parameters, and network binarization in algorithm-based approaches [4], [5]. A recent development is reducing computing complexity and model size by using low-bit-width weights and activations. By converting the Multiplication-And-Accumulate (MAC) operation into the corresponding AND-bitcount operations in [4], the authors performed bit-wise convolution between the inputs and the low-bit-width weights. Binarized convolutional neural networks, as an extreme quantization method [3], have achieved acceptable accuracy on both small [6] and large datasets [5] by removing some high precision requirements. By binarizing the weight and/or input feature map, they offer a promising solution to mitigate the aforementioned bottlenecks in storage and computation.

From the hardware point of view, the underlying operations should be realized using efficient mechanisms. The conventional processing elements are designed to work with a von-Neumann computing model involving separate memory and processing blocks interconnected via buses, which poses serious problems, such as long memory access latency, limited memory bandwidth, energy-hungry data transfer, and high leakage power consumption, which limit the edge device's efficiency and working time [2]. Additionally, this presents several major issues at the upper level, including bandwidth congestion and security concerns. The concept of instant image pre-processing with smart image sensors has therefore been extensively investigated [2], [7]–[9] as a potential remedy. Using an on-chip processor, pixels' digital outputs can be accelerated where the sensor is located, paving the way for enhanced sensor paradigms such as Processing-Near-Sensor (PNS). Other promising alternatives are a Process-in-Sensor (PIS) platform [8], [10] that processes pre-Analog-to-Digital Converter data and a hybrid PIS-PNS [1] platform to incorporate vision sensors and eliminate redundant data output. Generally, PIS units process images before transmitting them to an on-chip processor for further processing. Typical designs rely on this type of data transfer (from CMOS image sensors to memory), which reduces the speed of feature extraction. With this PIS unit, a computation core can (i) significantly reduce the power consumption of converting photo-currents into pixel values used for image processing, (ii) accelerate data processing, and (iii) alleviate the memory bottleneck problem [1], [2].

In this paper, we propose a novel low-power processing in-sensor scheme with event and object detection capabilities to alleviate power costs of data conversion and transmission, namely, TizBin. It offers several unique features, such as analog convolutions enabling low-precision ternary weight neural networks (TWNN) to mitigate the overhead of analog buffer and analog-to-digital converters. Once the moving object is detected, TizBin, as an always-on intelligent visual perception architecture, switches to the high-power object detection mode to capture the image.

2. Background

Sensors that detect a field of view are responsible for generating a stream of pixels representing the scenic event

2576-6996/22/$31.00 ©2022 IEEE
DOI 10.1109/ICCD56317.2022.00117

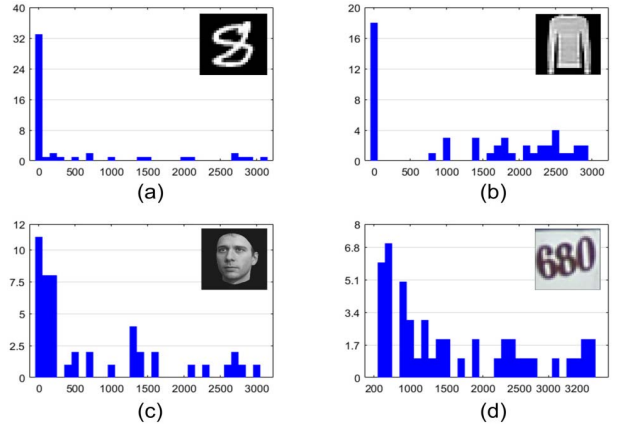


Figure 1: Distribution of mean absolute deviation where image is divided into 5×5 patches for (a) MNIST, (b) FashionMNIST, (c) MCFD, and (d) SVHN datasets.

for a backend processor, which is analogous to the function of the eye-brain system. The human eye's retina contains 130 million pixels, with only 1.3 million synaptic connections to the brain, indicating a high sparsity ratio. This massive sparsity is key to minimizing power consumption and latency, as well as preventing redundant information from being sent to the brain. The correlation between raw images within local regions tends to be positive, and a small amount of data is passed on to the higher processing stages. In the biological vision system, early sensory processing reduces redundancy based on information-theoretical considerations [11]. It is argued that this model explains how the visual system eliminates redundancy by predicting incoming signals and has an internal model of how to eliminate these signals [12]. It has been determined that if an image frame is divided into logical regions, the homogeneous ones comprise the majority of the redundancies. A measurement of variation enables the identification of these regions. To do so, the following expressions are utilized, where the mean absolute deviation (MAD) is the hardware-friendly and approximated version, and μ is the mean value.

$$\sigma^2 = \frac{\Sigma_{i=1}^{n}(x_i - \mu)^2}{n}, \quad MAD = \frac{\Sigma_{i=1}^{n}|x_i - \mu|}{n} \tag{1}$$

Figure 2 depicts the data distribution in MNIST, FashionMNIST, MCFD, and SVHN. For instance, in the three first datasets (a)-(c), there are many regions with zero value. SVHN (d), however, has fewer homogeneous regions due to the higher background to foreground pixel ratio.

2.1. Near/In -Sensor Processing Background

Systematic integration of computing and sensor arrays has been widely studied to eliminate off-chip data transmission and reduce ADC bandwidth, known as a processing-near-sensor (PNS) [9], [13], combining sensor and processing element so-called processing-in-sensor (PIS) [14], [15], [25], [26], and finally integrating pixels and computation unit, known as a processing-in-pixel (PIP) [8], [9]. In [9], photocurrents are converted into pulse-width modulation signals, and a dedicated analog processor is used to perform

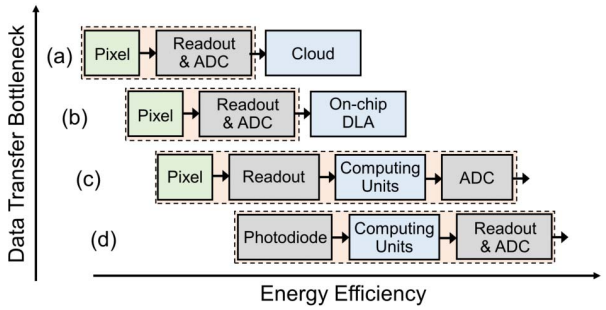


Figure 2: Visual systems with different architectures; (a) Conventional architecture, (b) PNS architecture, (c) PIS architecture, and (d) PIP architecture, where green and orange boxes indicate the pixel and the sensors. respectively, and blue boxes represents where the computing are performed.

feature extraction, reducing the amount of power consumed by the ADC. To run spatiotemporal image processing, 3D-stacked column-parallel ADCs and processing elements are implemented and utilized in [2]. The CMOS image sensor with dual-mode delta-sigma ADCs described in [16] is designed to process 1^{st}-conv. layer of binarized-weight neural networks (BWNN). Charge-sharing tunable capacitors are used by RedEye [17] to implement the convolution operation. By sacrificing accuracy in favor of energy savings, this design reduces energy consumption compared to a CPU/GPU. However, for high accuracy computation, the required energy per frame increases dramatically by $100\times$. As a PIS platform, MACSen [8] processes the 1^{st}-conv. layer of BWNNs with the correlated double sampling procedure and achieves speeds of 1000fps in computation mode. This method, due primarily to the SRAM-based PIS, however, suffers from a humongous area overhead and high power consumption. An example of a pulse-domain algorithm is [18], which optimizes near-sensor image processing by using photodiode arrays and an ADC to minimize design complexity and increase cost and speed.

Similar to the previous state-of-the-art works, we mainly focus on the first layer for the following reasons and observations. From the accuracy point of view, in the most quantized neural network accelerators, the first and the last layers of the networks remain in the full-precision, floating-point domain, while in [19], authors showed that in vision-based applications, the input feature map generally includes fewer channels (e.g., red, green, and blue) compared to the internal layers (e.g., 512). Thus, the first convolution layer often has the least computation [20], while communications are relatively high. Besides, continuous-valued inputs can be easily handled as fixed points with n bits of precision. An example of a 4-bit fixed point input would be $s = x.w^b$; $s = \sum_{n=1}^{4} 2^{n-1}(x^n.w^b)$, where x_1^4 is the most significant bit of the first input, w is 1-bit weights, and a is a 4-bit input. From the efficiency perspective, because raw image data is full precision, the first layer's convolution operations are the performance bottleneck in different hardware/software co-design accelerators and require a lot of memory and processing resources [21]. We used the deep neural network energy estimation tool developed by

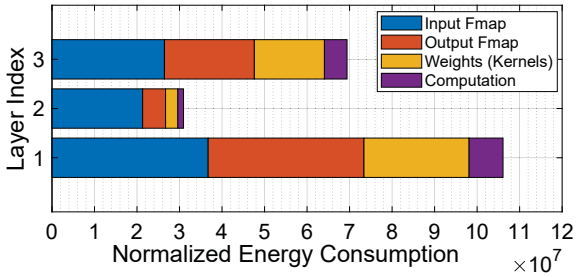


Figure 3: Energy consumption for a 3-layer MLP network.

Yang et al. [22] to demonstrate that the first layer of our target network consumes more power. Figure 3 depicts the breakdown of normalized energy consumption according to layers. As observed, layer one consumes much more energy than the other layers for computation (purple block) and data movement (the other three blocks). On the other hand, in conventional image sensors: most of the power (>96% [23]) is consumed by processing and converting pixel values. That means pixel circuits consume only 4 percent of power to perform photovoltaic conversions, whereas signal amplification, digital-to-analog conversion, and data transmission consume most of the power; finally, almost all the PNS/PIS/PIP systems are hardwired, so the functionalities are limited to simple pre-processing tasks such as 1^{st}-layer BWNN computation.

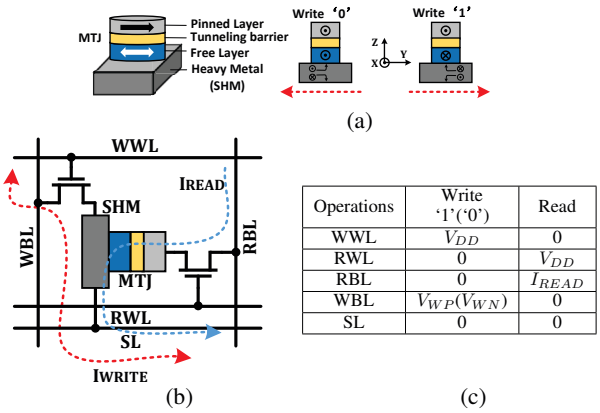
2.2. SOT-MRAM

Figure 4a shows a Spin-Orbit Torque Magnetic Random Access Memory (SOT-MRAM) device structure used in this work. The storage element in SOT-MRAM is SHE-MTJ [24], a composite device structure of a Spin Hall Metal (SHM) and Magnetic Tunnel Junction (MTJ). The binary data is stored as resistance states of MTJ. Data-'0'(/'1') is encoded as the MTJ's lower(/higher) resistance or parallel(/anti-parallel) magnetization in both magnetic layers (free and fixed layers). Here the flow of charge current ($\pm$y) through the SHM will cause accumulation of opposite directed spin on both surfaces of SHM due to spin Hall effect [24]. Thus, a spin current flowing in $\pm$z is generated and further produces spin-orbit torque (SOT) on the adjacent free magnetic layer, causing a switch of magnetization. Each cell located in the computational sub-array is connected with a Write Word Line (WWL), Write Bit Line (WBL), Read Word Line (RWL), Read Bit Line (RBL), and Source Line (SL). The bit-cell structure of 2T1R SOT-MRAM and its biasing conditions are shown in Fig. 4b and 4c, respectively. In this work, the magnetization dynamics of Free Layer ($\boldsymbol{m}$) are modeled by LLG equation with spin-transfer torque terms, which can be mathematically described as [24]:

$$\frac{d\boldsymbol{m}}{dt} = -|\gamma|\boldsymbol{m}\times\boldsymbol{H_{eff}} + \alpha\left(\boldsymbol{m}\times\frac{d\boldsymbol{m}}{dt}\right) + |\gamma|\beta(\boldsymbol{m}\times\boldsymbol{m}_p\times\boldsymbol{m}) - |\gamma|\beta\epsilon'(\boldsymbol{m}\times\boldsymbol{m}_p) \quad (2)$$

$$\beta = |\frac{\hbar}{2\mu_0 e}|\frac{I_c P}{A_{MTJ}t_{FL}M_s} \quad (3)$$

where $\hbar$ is the reduced plank constant, γ is the gyromagnetic ratio, I_c is the charge current flowing through MTJ, t_{FL} is



Operations	Write '1'('0')	Read
WWL	V_{DD}	0
RWL	0	V_{DD}
RBL	0	I_{READ}
WBL	$V_{WP}(V_{WN})$	0
SL	0	0

(c)

Figure 4: (a) SOT-MRAM device structure, (b) Schematic and (c) biasing conditions of SOT-MRAM bit-cell.

the thickness of free layer, ϵ' is the second Spin transfer torque coefficient, and $\boldsymbol{H}_{\text{eff}}$ is the effective magnetic field, P is the effective polarization factor, A_{MTJ} is the cross-sectional area of MTJ, $\boldsymbol{m_p}$ is the unit polarization direction. Note that the ferromagnets in MTJ have In-plane Magnetic Anisotropy (IMA) in x-axis [24]. With the given thickness (1.2nm) of the tunneling layer (MgO), the Tunnel Magneto-Resistance (TMR) of the MTJ is $\sim$ 171.2%.

3. TizBin Architecture

We propose TizBin as an efficient and reconfigurable event and object detection sensor architecture to address the aforementioned challenges and limitations. TizBin consists of an $m \times n$ Compute Focal Plane (CFP) array, row and column controllers (Ctrl), command decoder, sensor timing Ctrl, a memory/computing unit, and readout/ADC/SA circuitry. The sensors operates in three modes, i.e., *sensing*, *event detection*, and *object detection* modes. The CFP is designed to co-integrate sensing and processing for low-power but high classification accuracy image processing applications. The output, i.e., preprocessed layer, is transmitted to an on-chip deep learning accelerator to accelerate further. Designing a domain-specific accelerator is out of scope.

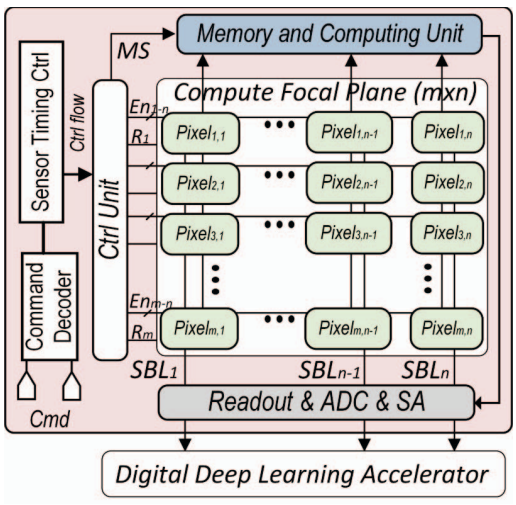


Figure 5: Proposed TizBin architecture.

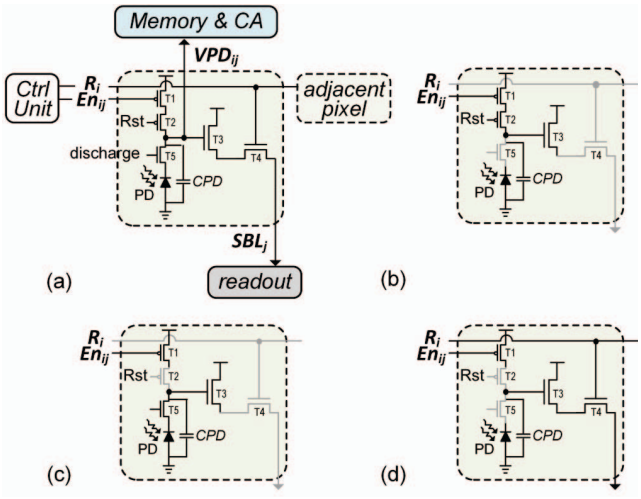


Figure 6: (a) The proposed sensor design, and its various phases (b) pre-charge, (c) evaluation, and (d) reading.

3.1. Proposed Pixel and Its Phases

The proposed pixel, as shown in Fig. 6a has 2 inputs and 2 outputs, which comprises 5 transistors, one Photodiode (PD), and one capacitor. There are three essential phases in pixels: *pre-charge*, *evaluation*, and *reading*. In the pre-charge phase, C_{PD} capacitor will charge to the V_{DD} value, and then, in the evaluating phase, C_{PD} will be discharged based on the resistance of the PD (based on light intensity). There are two design considerations and novelties, comparing the proposed pixel design to previous works: (1) the C_{PD} capacitor will only be discharged in the evaluation phase due to the T5 transistor. As a result, in the reading phase, the pixel value remains unchanged; therefore, we do not need to charge C_{PD} from '0,' decreasing overall power consumption; (2) The T1 transistor is added to turn off the pixel using the control unit. By turning off this transistor, C_{PD} is never charged, providing reconfigurability for the pixel and saving more power.

In the *pre-charge* phase, Rst = '0' therefore, if En = '0' then CPD will be charged to V_{DD} through T1 and T2; otherwise, T1 is off, and there is no path to charge CPD. This is how we turn off the pixels. In this phase, discharge = '0'; therefore, T5 is off (Fig. 6b). In the *evaluation* phase (Fig. 6c), after charging CPD, both Rst and discharge signals change to '1'. In this step, based on the light's density, the photodiode's resistance will change; as a result, different values of CPD will be discharged through the T5. In the *reading* phase, as shown in Fig. 6d, values of the pixels are read row by row based on the R_i signal. In this phase, Rst = '1 and discharge = '0'; as a result discharging process will stop through T5, and the value of the CPD will stay constant. When row i^{th} is selected using the control unit, T4 will be turned on, and T3 creates a current based on the CPD voltage on its gate. This current will convert to the voltage on the SBL line and will be measured using the readout circuit at the end of each SBL. Since pixels

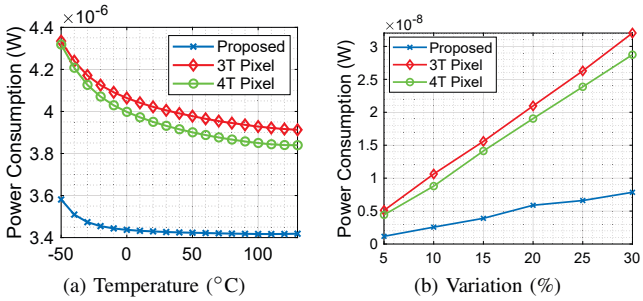


Figure 7: Relationship between the power consumption and two types of factors, (a) temperature, and (b) mismatch.

are read in a row-by-row manner, it takes a while. Thus, in 3T and 4T -pixel designs, values of the lower pixels are changed, resulting in more power consumption than the proposed pixel. In our architecture, T5 prevents undesirable discharging and saves more power while generating a more explicit photo. The proposed pixel simulates various situations, including temperature and mismatching of both capacitor and transistor sizes. As shown in Fig. 7a and 7b the proposed design is more resilient in both situations.

3.2. Proposed Peripheral Circuits

Compute Add-on (CA): In addition to the pixel array, our architecture includes a Memory and Computing Unit (Fig. 8a). The memory unit consists of $M \times N$ SOT-MRAM (Fig. 8b) and one 1D Compute Add-on array (Fig. 8c). The memory unit functions similarly to a conventional memory component. While TizBin uses an efficient compute add-on array to support the computation required for event- and object-detection tasks. This compute array comprises $M \times N$ CAs, each connected to a SOT-MRAM within the memory unit and controlled by the voltage value of each pixel (V_{PD}). The structure of the CA is shown in Fig. 8c. The CA array is designed to generate positive and negative current flow on the Current Bit Line (CBL) based on the value stored in the memory. Due to the particular structure of the proposed pixel, the proposed CA can produce positive

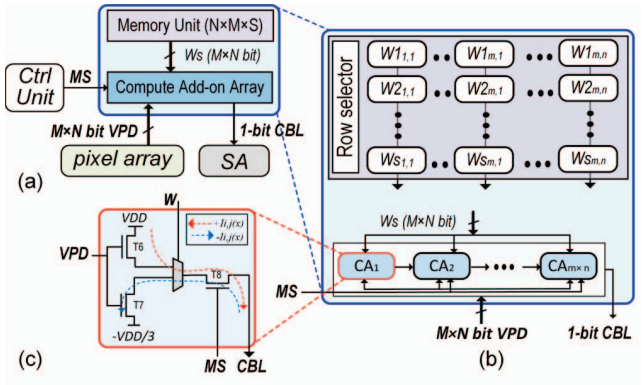


Figure 8: The proposed Memory and Computing Unit. (a) Its structure w.r.t. inputs and output. (b) $M \times N$ SOT-MRAM cells and $1 \times (M \times N)$ CA arrays, and (c) a CA design.

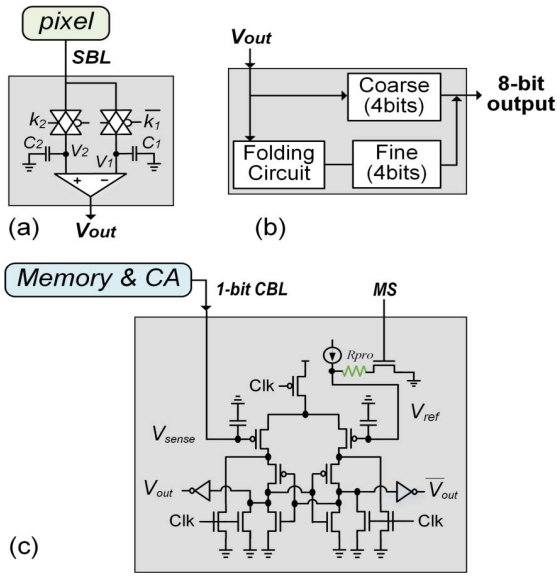


Figure 9: The proposed circuits of (a) readout, (b) folding ADC, and (c) sense amplifier.

and negative currents, in addition to zero current, by turning off the sensor. This way, ternary weights of the DNN can be stored and processed for the object-detection task.

Readout and ADC: Readout and ADC circuits are shown on Fig. 9a and Fig. 9b, respectively. These circuits are used in the sensing and event detection modes. In the readout circuit (Fig. 9a), the first value of the pixel is stored on the C1 using transmission gate k_1, and in the next step, after reducing the value of CPD, the new value of the pixel will be store on the capacitor C2. A subtractor connected to these capacitors subtracts the old and new values of the pixel. This value will be measured using the 8-bit ADC. In the Tizbin structure, we use folding ADC (Fig. 9b) rather than flash ADC. The folding ADC consists of two parts, coarse and fine. The coarse circuit is responsible for the most significant bits (MSBs), while the fine part generates 4 least significant bits (LSBs). For an 8-Bit flash ADC, we need 256 comparators, while in a folding ADC, we need only 32 comparators. In the sensing mode, we need all 8 bits, while in event-detection mode, only 4 MSBs are required. Therefore, TizBin turns off the folding and fine circuit to save more power and memory.

Sense Amplifier: Transistor-level schematic of the used sense amplifier is presented in Fig. 9c. As illustrated in the figure, SA considers three inputs named CLK, mode selector (MS), and CBL. The functionality of this sense amplifier is like a sign function. It means that for voltage bigger than Vref, the output of this circuit will be one; otherwise, the result is zero. It should be mentioned the output of SA is valid when CLK = '0' and MS = '1'.

3.3. Putting All Together

The proposed TizBin comprises 600×600 pixels. We architect the pixels in the groups of $5{\times}5$[1] called Box as shown in Fig. 10a. Each pixel box is then divided into two parts based on its position. The central pixels (i.e., *PVT*, Fig. 10b) are dedicated to participating in both event and object detection modes, whereas the rest of the pixels are used only for object detection. TizBin offers three main modes, including *Sensing*, *Event-Detection* and *Object-Detection* modes, which are chosen by a MS signal.

3.3.1. Sensing Mode. Reading the pixels' values in the sensing mode is performed in a row-by-row manner; therefore, reading all pixels requires r clock cycles, where r is the number of rows. Initially setting Rst='high' in the sensing mode in Fig. 6a the PD connected to the T2 transistor turns into inverse polarization. Turning on the access transistor T4 and k_1 switch at the shared ADC (Fig. 9a) allows the C_1 capacitor to charge through SBL fully. By turning off T2, PD generates a photo-current concerning the external light intensity, which leads to a voltage drop (V_{PD}) at the gate of T3. Once again, by turning on T4 and this time k_2 switch, C_2 is selected to record the voltage drop. Therefore, the voltage values before and after the image light exposure, i.e., V_1 and V_2 in Fig. 9a, are sampled. The difference between two voltages is sensed with an amplifier, while this value is proportional to the voltage drop on V_{PD}. In other words, the voltage at the cathode of PD can be read at the pixel output.

3.3.2. Event-Detection Mode. The operation principle of the object-detection mode is explained in three steps, i.e., read, calculation (compare), and box activation, presented using Algorithm 1. In the reading step (line 4), only the PVT of each box is turned on, and the remaining pixels are disconnected from the power supply. For example, the original raw image and only activated PVTs are depicted in Fig. 11a and Fig 11c, respectively. If we concatenate these central pixels, Fig 11e is generated, which comprises only 120×120 pixels rather than 600×600 pixels. In the calculation step (lines 6-8 of Algorithm 1), the value of each PVT is measured like the sensing mode. Nonetheless, in this step, TizBin does not need to use all 8-bits of ADC, and only 4 bits of each central pixel will be measured and compared with the value of the pixels in the previous step leveraging the proposed ADC in Fig. 9. In the precise sensing, if two values are equal, *interpreted as the inactivity*, other pixels in the box remain inactive. In the case, in-quality, *interpreted as the activity* (line 7 of Algorithm 1), represents as a region of interest. In the next Clk cycle, all pixels of these boxes

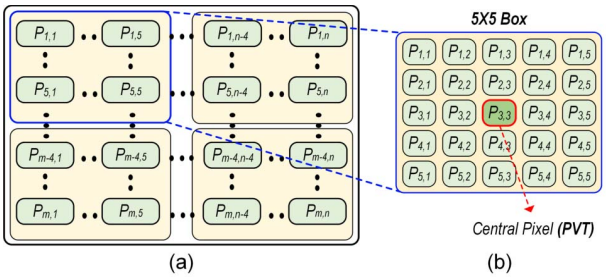


Figure 10: (a) An $M \times N$ pixel array, including $M/5 \times N/5$ boxes. (b) Box 1, consisting of 25 pixels and its central pixel.

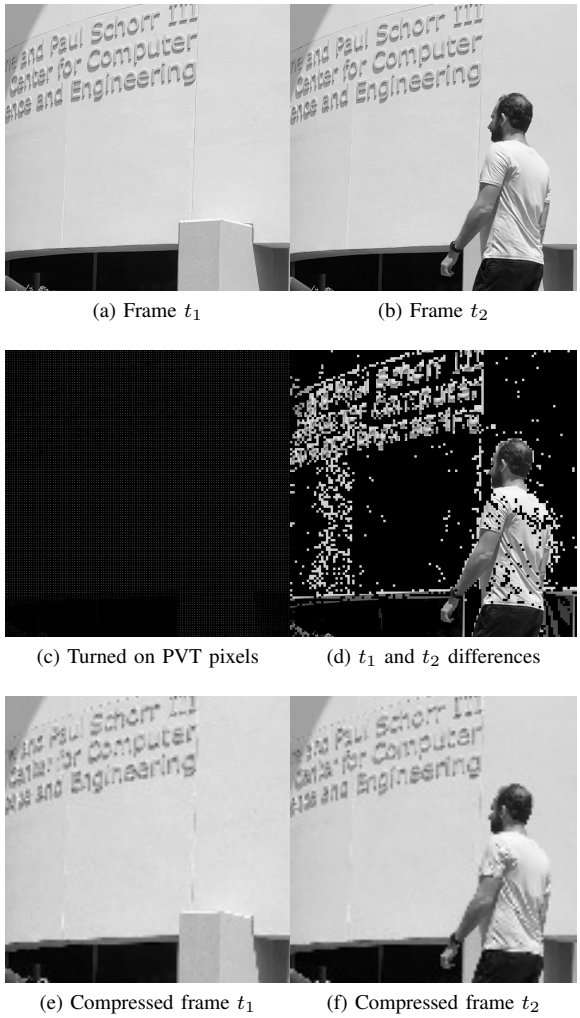
1. There are 120 boxes with the size of 5×5.

Algorithm 1 In-Sensor Event-Detection Algorithm

```
1: Input: 600 × 600 pixel array
2: Output: Activated Boxes
3: procedure ISED
4:     pixel_values ← Read (central_pixels)
5:     turn_on_list = []
6:     for i ← 0 to |pixel_values| do
7:         if pixel_values [i] != old_pixel_values [i] then
8:             turn_on_list.Push (pixel_values [i])
9:     old_pixel_values = pixel_values
10:    while (length (turn_on_list) !=0) do
11:        turn_box_on (turn_on_list.Pop)
```

become active by the control unit, and the new value of the sensor will be stored in the memory (lines 10 and 11). The turned-on boxes are shown in Fig 11d. As result, by changing the mode to object-detection, the results can be processed by the CNN algorithm with only these areas and then system turns to the detection mode again.

3.3.3. Object-Detection Mode. In this mode, the C_{PD} capacitor is initialized to the fully-charged state by setting Rst='high', similar to the sensing mode. During an evaluation cycle, by turning off T1, the Ctrl Unit activates the CBL signal, while the R_i signals are deactivated. This will activate the entire array for a single-cycle MAC operation. The core idea behind compute add-on shown in Fig. 8c is to leverage pixel's V_{PD} as a sampling voltage for T6(/T7) in v-SOT-MRAMs to generate(/pull) current from the CBLs simultaneously. To implement multiplications between the pixel value identified by V_{PD} and the binary weight stored in SOT-MRAM, a 2:1 MUX unit was devised in every CA, taking the T6's source and T7's drain signals as inputs and the SOT-MRAM sensed data as the selector. Note that T6 and T7 are connected to V_{DD} and $\frac{-V_{DD}}{3}$, respectively. After exposure, the set of input sensor voltages V_{PD}= [$V_{PD_{1,1}}$, $V_{PD_{1,2}}$,..., $V_{PD_{m,n}}$] is applied to the gate of T6s and T7s generating current set I= [$I_{1,1(1)}$, $I_{1,1(2)}$,..., $I_{1,1(v)}$,..., $I_{m,n(1)}$, $I_{m,n(2)}$,..., $I_{m,n(v)}$] for the entire array. If the binary weight equals '1' (*Wi*=+1), T6 acts as a current source and generates a current with $I_{i,j(x)}$ magnitude on the shared CBL as shown by the red dashed line in Fig. 8c. However, if the binary weight equals '0' (*Wi*=-1), T5 transistor acts as a negative current source and pulls a current with the same magnitude as $I_{i,j(x)}$ in the opposite direction from the shared CBL as indicated by the blue dashed line in Fig. 8c. Please note that T6's and T7's gate capacitors as well as parasitic capacitors will be fully charged to V_{DD} through T1 in the pre-charge cycle, this will significantly keep the pixel sensitivity when the number of compute add-ons increases. Mathematically, let $G_{j,i}$ be the conductance of the synapse connecting i^{th} to the j^{th} node, the current through that synapse is $G_{j,i}V_i$ and the collection of the current through each CBL represents the MAC result ($I_{sum,j}$=$\sum_i G_{j,i}V_i$), according to Kirchhoff's law. This is readily calculated by measuring the voltage across a sensing resistor. This mechanism converts every input pixel value to a weighted current according to the SOT-MRAM that is interpreted as the multiplication in DNNs. For the activation function, we



(a) Frame t_1 (b) Frame t_2

(c) Turned on PVT pixels (d) t_1 and t_2 differences

(e) Compressed frame t_1 (f) Compressed frame t_2

Figure 11: (a) and (b) are two different frames. (c) Only PVTs are on. (d) Differences between two frames. (e) and (f) show when only PVT pixels are connected.

designed and tuned a sense circuit connected to each CBL based on StrongARM latch to realize an in-sensor $sign$ function [19] as shown in Fig. 9c. The sense amplifier requires two clock phases: pre-charge (Clk 'high') and sensing (Clk 'low'). During sensing, $I_{sum(x)}$ flows from every CBL to the ground and generates a sense voltage (V_{sense}) at the input of the sense amplifier. This voltage is compared with the reference voltage by applying a proportional current over a processing reference resistor (R_{pro}) activated by the mode signal. The binary activation is then transmitted through the bus fabrics to the PNS unit for storage.

4. Simulation Results

Functionality: Figure 12 shows the functionality of one proposed pixel. When En equals V_{DD}, VPD is never charged, and the produced current on CBL approximately is zero. On the other hand, by changing the EN value to zero, with the first Rst clock ('0'), CPD is charged to V_{DD}, and when Rst

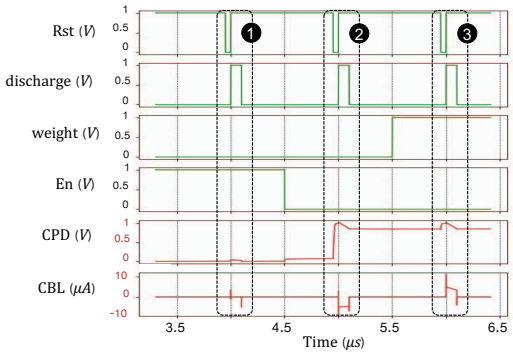


Figure 12: Transient simulation waveform of a pixel with a single CA.

is returned to '1' again, and $discharge$ is V_{DD}, CPD starts discharging. In the end, the value of the VPD has remained the same. Everything in this step is similar to the previous one except the weight values. Before starting the sensing and processing phases, the pre-trained weights should be written into NVMs and remain unchanged in the proper memory unit. Nevertheless, to evaluate the output current, we changed the pixel weights. This simulation indicates sensor with negative and positive weights produces a current value of approximately $-5\mu A$ and $+5\mu A$, respectively.

The transient simulation results of an 8×1 pixel array are shown in Fig. 13. Herein, eight sensors are connected to the CBL. The results are obtained in the presence of 15% process variation in transistor sizing for 1000 simulation runs. To verify sensors' functionalities, this evaluation phase can be divided into two steps. In the first step, some sensors were disabled. Therefore, the sum of currents according to their weights becomes approximately $-1\mu A$ at the rising edge of the Clk signal. As previously mentioned, the current value smaller than 0 interprets as '0', and bigger than 0 denotes as '1'. Therefore, the output of the SA (out) is '0', whereas, in the second step, the weights changed and generated a positive current, and out became 1. As depicted in Fig. 13, the proposed pixel is resilient during the process variation, and all waveforms approximately have the same value in each iteration.

Performance Evaluation: Table 1 compares the structural and performance parameters of selective PIS and PIP designs in the literature. As different designs are developed

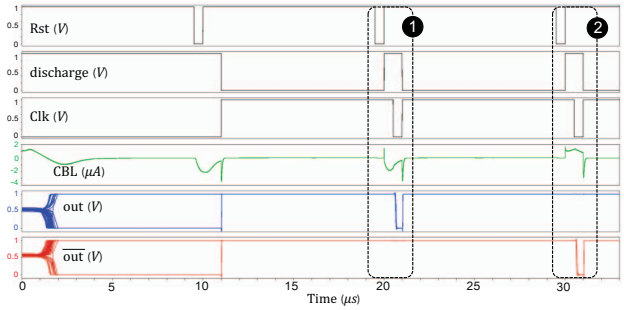


Figure 13: Transient simulation waveform of an 8×1 sensor array.

TABLE 1: Performance comparison of various sensor units.

Designs	Technology (nm)	Purpose	Frame Rate ($frame/s$)	Power (mW)	Efficiency ($TOp/s/W$)
[27]	180	2D optic flow est.	30	0.029	0.0041
[9]	180	edge*/blur/sharpen/ 1st layer DNN	480	sensing: 0.077 processing: 0.091	0.777
[2]	60/90	STP†	1000	sensing: 230 processing:363	0.386
[8]	180	1st layer BNN	1000	0.0121	1.32
[7]	180	edge*/TMF‡	100,000	1230	0.535
TizBin	65	edge 1st layer DNN	1000	sensing: 0.025 processing: 0.0088	1.83

* Edge extraction. † Spatial Temporal Processing. ‡ Thresholding Median Filter.

for specific domains, for an impartial comparison, we estimated and normalized the power consumption when all PIS units execute the similar task of processing the 1^{st}-layer of DNN. The TizBin achieves the frame rate of 1000 and the efficiency of ∼1.83 TOp/s/W as the most efficient design. However, the structure in [7] achieves the highest frame rate. As we do not have access to the other layouts' configurations, it is challenging to have a fair comparison between area overheads. However, we believe a ballpark assessment can be made by comparing the number of transistors in previous SRAM-based designs and TiZBin's lower-overhead compute add-on.

4.1. High-level Evaluation

We demonstrate the advantages of TizBin design through an image classification task. In the original BWNN and TWNN topologies, all the layers, except the first and last, are implemented with quantized weights [19], [28], [29]. Since, in image classification tasks, the number of input channels is relatively smaller than the number of internal layers' channels, the required parameters and computations are small. Thus, converting the input layer will not be a significant issue [19]. Three PIP designs, including 3T and 4T -pixel designs, and our TizBin are considered. The first two architectures can implement BWNNs (-1, +1), while our pixel implements TWNN (-1, 0, +1). After performing the first layer's computations, the remaining layers can be accelerated with an identical NN accelerator. To do so, the outputs of the 1^{st}-layer are then fed into the second layer of the algorithm, which is implemented in Python.

NN Architecture: In order to evaluate our design and perform a fair comparison, we developed a 2-layer MLP with 1024 inputs, 16 hidden nodes, and 10 outputs.

Datasets: We conduct experiments on several datasets to evaluate the performance of TizBin, including MNIST [30], Fashion-MNIST [31], MCFD [32] and SVHN [33]. MNIST is leveraged as a gray-scale dataset that contains 70,000 28×28 images of handwritten digits from 0 to 9, 60,000 images for training, and 10,000 images for testing sets. Similar to MNIST, Fashion-MNIST consists of 28×28 gray-scale images but includes 10,000 images for each training and testing set to form ten fashion categories. MCFD face recognition database contains face images of 10 subjects, where each image is normalized to 20×20 pixels. Training data consists of 6,977 images, while testing data consists of 24,045 images. Finally, we also exploit SVHN with 73257 training digits, 26032 testing digits, and 531131 additional digits for extra training data. The images are pre-processed

to 20×20 from the original 32×32 cropped version and fed to the model.

Accuracy: We conduct experiments on the mentioned datasets. The comparison of classification accuracy is summarized in Table 2. The results show that our TizBin architecture provides higher accuracy can be achieved rather than BWNNs based on 3T and 4T -pixels. This improvement is because of three values realized by the proposed pixel. It is worthy to note that we can alter BWNNs' values from (-1, +1) to (0, +1), which causes several issues like no guarantee for convergence.

TABLE 2: Classification accuracy (%) on MNIST, Fashion-MNIST, MCFD and SVHN.

Configuration	MNIST	FashionMNIST	MCFD	SVHN
BNN [19]	98.6	90.02	–	97.47
PIP [8]	96.0	83.17	90.67	–
TizBin	97.38	85.68	92.30	91.05

*Binarized Neural Network (BNN) and PIP are a software and a hardware -based implementations, respectively. It worthy noted that, PIP can be implemented using 4T-pixel and 3T-pixel, while both provide the same accuracy.

5. Conclusion

This paper proposed TizBin, as an always-on intelligent visual perception architecture that realizes a low-power processing in-sensor scheme with event and object detection capabilities. TizBin supports analog convolutions enabling low-precision TWNN to mitigate the overhead of analog buffer and analog-to-digital converters. Once the moving object is detected, it switches to the high-power object detection mode to capture the image. Our results demonstrate acceptable accuracy on various data sets, while TizBin achieves a frame rate of 1000 and efficiency of ∼1.83 TOp/s/W.

Acknowledgements

This work is supported in part by the National Science Foundation under Grant No. 2216772 and 2216773.

References

[1] T.-H. Hsu *et al.*, "Ai edge devices using computing-in-memory and processing-in-sensor: from system to device," in *2019 IEDM*. IEEE, 2019, pp. 22–5.

[2] T. Yamazaki *et al.*, "4.9 a 1ms high-speed vision chip with 3d-stacked 140gops column-parallel pes for spatio-temporal image processing," in *ISSCC*. IEEE, 2017, pp. 82–83.

[3] J. H. Ko *et al.*, "A single-chip image sensor node with energy harvesting from a cmos pixel array," *IEEE TCASI*, vol. 64, 2017.

[4] S. Zhou *et al.*, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv:1606.06160*, 2016.

[5] M. Rastegari *et al.*, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.

[6] C. Matthieu et al., "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[7] S. J. Carey *et al.*, "A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array," in *2013 Symposium on VLSI Circuits*. IEEE, 2013, pp. C182–C183.

[8] H. Xu *et al.*, "Macsen: A processing-in-sensor architecture integrating mac operations into image sensor for ultra-low-power bnn-based intelligent visual perception," *IEEE TCAS II*, vol. 68, pp. 627, 2020.

[9] T.-H. Hsu *et al.*, "A 0.5-v real-time computational cmos image sensor with programmable kernel for feature extraction," *IEEE JSSC*, vol. 56, pp. 1588–1596, 2020.

[10] H. Xu *et al.*, "Senputing: An ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing," *IEEE TCASI*, 2021.

[11] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, "Predicting human gaze beyond pixels," *Journal of vision*, vol. 14, no. 1, pp. 28–28, 2014.

[12] Y. Huang and R. P. Rao, "Predictive coding," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 5, pp. 580–593, 2011.

[13] Q. Li *et al.*, "Ns-fdn: Near-sensor processing architecture of feature-configurable distributed network for beyond-real-time always-on keyword spotting," *IEEE TCASI*, vol. 68, no. 5, pp. 1892–1905, 2021.

[14] H. Xu *et al.*, "Utilizing direct photocurrent computation and 2d kernel scheduling to improve in-sensor-processing efficiency," in *DAC*. IEEE, 2020, pp. 1–6.

[15] H. Xu *et al.*, "A 4.57 μw@ 120fps vision system of sensing with computing for bnn-based perception applications," in *A-SSCC*. IEEE, 2021, pp. 1–3.

[16] W.-T. Kim *et al.*, "An on-chip binary-weight convolution cmos image sensor for neural networks," *IEEE TIE*, vol. 68, pp. 7567–7576, 2020.

[17] R. LiKamWa *et al.*, "Redeye: analog convnet image sensor architecture for continuous mobile vision," *ACM SIGARCH Computer Architecture News*, vol. 44, pp. 255–266, 2016.

[18] F. Taherian and D. Asemani, "Design and implementation of digital image processing techniques in pulse-domain," in *APCCAS*. IEEE, 2010, pp. 895–898.

[19] I. Hubara *et al.*, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[20] C. Szegedy *et al.*, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[21] F. Muñoz-Martínez *et al.*, "Stonne: Enabling cycle-level microarchitectural simulation for dnn inference accelerators," in *IEEE IISWC*, 2021, pp. 201–213.

[22] T.-J. Yang *et al.*, "A method to estimate the energy consumption of deep neural networks," in *2017 51st asilomar conference on signals, systems, and computers*. IEEE, 2017, pp. 1916–1920.

[23] J. Choi *et al.*, "An energy/illumination-adaptive cmos image sensor with reconfigurable modes of operations," *IEEE JSSC*, vol. 50, pp. 1438–1450, 2015.

[24] X. Fong *et al.*, "Spin-transfer torque devices for logic and memory: Prospects and perspectives," *IEEE TCAD*, vol. 35, pp. 1–22, 2015.

[25] S. Angizi *et al.*, "Pisa: A binary-weight processing-in-sensor accelerator for edge image processing," *arXiv preprint arXiv:2202.09035*, 2022.

[26] M. Abedin *et al.*, "MR-PIPA: An Integrated Multi-level RRAM (HfOx) based Processing-In-Pixel Accelerator," *IEEE JXCDC*, 2022.

[27] S. Park *et al.*, "7.2 243.3 pj/pixel bio-inspired time-stamp-based 2d optic flow sensor for artificial compound eyes," in *ISSCC*. IEEE, 2014, pp. 126–127.

[28] M. Ghasemzadeh *et al.*, "Rebnet: Residual binarized neural network," in *FCCM*. IEEE, 2018, pp. 57–64.

[29] Y. Umuroglu *et al.*, "Finn: A framework for fast, scalable binarized neural network inference," in *FPGA*, 2017, pp. 65–74.

[30] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[31] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[32] C. for Biological, C. L. at MIT, and MIT. (2000) Cbcl face fatabase. [Online]. Available: http://cbcl.mit.edu/software-datasets/FaceData2.html

[33] Y. Netzer *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS*, vol. 2011, no. 2, 2011, p. 5.