An Ultra-Low-Power Image Signal Processor for Hierarchical Image Recognition With Deep Neural Networks

Hyochan An, *Student Member, IEEE*, Sam Schiferl, Siddharth Venkatesan, Tim Wesley, Qirui Zhang, *Graduate Student Member, IEEE*, Jingcheng Wang, *Member, IEEE*, Kyojin D. Choo, *Member, IEEE*, Shiyu Liu, Bowen Liu, *Graduate Student Member, IEEE*, Ziyun Li, *Member, IEEE*, Luyao Gong, Hengfei Zhong, David Blaauw, *Fellow, IEEE*, Ronald Dreslinski, *Senior Member, IEEE*, Hun Seok Kim, *Member, IEEE*, and Dennis Sylvester, *Fellow, IEEE*

***Abstract*— We propose an ultra-low-power (ULP) image signal processor (ISP) that performs on-the-fly in-processing frame compression/decompression and hierarchical event recognition to exploit the temporal and spatial sparsity in an image sequence. This approach reduces energy consumption spent processing and transmitting unimportant image data to achieve a 16× imaging system energy gain in an intruder detection scenario. The ISP was fabricated in 40-nm CMOS and consumes only 170 μW at 5 frames/s for neural network-based intruder detection and 192× compressed image recording.**

***Index Terms*— Deep neural network (DNN), energy-efficient processor, event recognition, image compression, image signal processor (ISP).**

I. Introduction

THE Internet of Things (IoT) is ubiquitous in many applications, such as smart homes, smart cities, and smart agriculture [1]. Battery-operated millimeter (mm)-scale IoT devices are desired solutions for embedding sensors in physical spaces due to their wireless operation and tiny form factor [2]. Imaging is a highly desirable sensing modality in these devices as it offers key contextual information about a system's environment. However, imaging requires a large amount of energy and storage space, creating challenges for mm-size systems.

Prior IoT imaging systems suffer from energy and storage size problems due primarily to the following two reasons: 1) many computer vision processors, such as [3]–[6], process uncompressed images, which necessitates a large frame buffer, thereby increasing the chip size and leakage power and 2) since many real-time image signal processors (ISPs), such as [7], [8], lack scene understanding, they cannot distinguish useful information, and all frames must be transmitted regardless of their importance, incurring considerable storage and wireless communication energy costs.

We found an opportunity to save energy by catering a system's data-path to the event frequency of a certain environment. For example, surveillance cameras in smart homes and offices tend to capture redundant images, such as unchanged background scenes, moving pets, or family members for most of the operation time, as depicted in Fig. 1(a). Recognizing and discarding unimportant images early in the computational pipeline allow the system to avoid expending energy on processing and transferring unimportant data. Given the expensive wireless data transmission and off-chip storage of battery-operated IoT devices, reducing the amount of data transmission required to ensure that the necessary useful information is transferred could optimize system energy, as shown in Fig. 1(b).

Therefore, we propose an ultra-low-power (ULP) ISP designed for size-constrained intelligent edge devices, as shown in Fig. 1(c). First, to reduce the required size of storage for frames, we employ macroblock (MCB)-based scene change detection (CD) using a new *sparse census-transform* encoding and JPEG compressed memory for input images. The proposed scheme ensures that full uncompressed images are never stored in their entirety on-chip. This reduces the required SRAM size needed to store frames on the chip by 11.2× and the leakage power by 26.9×. Second, to understand the scene, we enable hierarchical event recognition through a programmable deep neural network (DNN) engine and a change detection engine (CDE), which progressively prunes uninteresting areas or the entire image. Since relevant information typically occurs sparsely in time and space, image storage and transmission requirements can be reduced by >1000×. Third, to reduce the size of storage required for algorithm parameters, DNNs use deep compression of all

Manuscript received August 30, 2020; revised November 4, 2020; accepted November 22, 2020. Date of publication December 14, 2020; date of current version March 26, 2021. This article was approved by Associate Editor Yusuke Oike. This work was supported by Sony Semiconductor Solutions Corporation/Sony Electronics Inc. *(Corresponding author: Hyochan An.)*

Hyochan An, Qirui Zhang, Kyojin D. Choo, Shiyu Liu, Bowen Liu, Hengfei Zhong, David Blaauw, Ronald Dreslinski, Hun Seok Kim, and Dennis Sylvester are with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: hyochan@umich.edu).

Sam Schiferl is with Amazon, Seattle, WA 98109 USA.

Siddharth Venkatesan is with Amazon, Santa Clara, CA 95054 USA.

Tim Wesley is with MemryX, Ann Arbor, MI 48105 USA.

Jingcheng Wang and H. Zhong are with Apple, Cupertino, CA 95014 USA.

Ziyun Li is with Facebook, Redmond, WA 98052 USA.

Luyao Gong is with Google, Mountain View, CA 94043 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2020.3041858.

Digital Object Identifier 10.1109/JSSC.2020.3041858

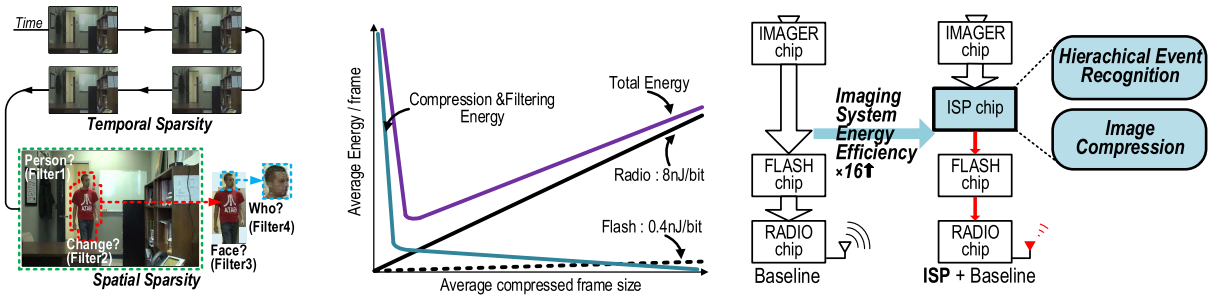


Fig. 1. Motivation for image processing intelligence at the edge embedded imaging systems. Taking advantage of sparsity in an image sequence can reduce the overall energy consumption of the embedded imaging system. (a) Sparsity in an image sequence. (b) Energy of an embedded imaging system. (c) Edge intelligence for an embedded imaging system.

on-chip weights stored in a custom ultra-low-leakage SRAM, further reducing the system size and power consumption. Fourth, an H.264 engine compresses the final detected regions-of-interest (RoIs), and the chip achieves a 192× total image size reduction ratio to reduce off-chip data transfer. In addition, all features of the ISP are highly flexible because it must adapt to the specific event frequencies of many different real-life environments.

The rest of this article is organized as follows. Section II introduces the overall architecture of the proposed ISP. Section III provides an example use scenario of the ISP. The three main innovations are illustrated in Sections IV–VI. Section VII shows the additional techniques used in the chip. The fabricated chip and measurement results are presented in Section VIII. Section IX concludes this article.

II. Architecture

Fig. 2 provides an overview of the top-level architecture of the proposed ISP design. The ISP consists of three customized IPs: an image streaming engine (ISE), a neural engine (NE), and an H.264 engine (H264E).

The ISE block processes streamed-in images on the fly. First, while a Bayer format image is streaming in from an imager [9], the pixels are calibrated using the optical black intensity of the front/back porch of the frame. The CDE performs customized MCB (16 × 16 pixels)-based CD on the calibrated image data. Following the operation, the IP performs de-Bayering and RGB-to-YUV converting on the changed MCBs. The MCBs are compressed into the JPEG compression memory. While each MCB has a variable length due to JPEG compression, they can be randomly accessed on demand.

The NE enables efficient DNN-based image recognition. The NE has a processing element (PE) to accelerate DNN operations. The NE control executor (NCX), a custom RISC processor, controls the operation of the PE by executing instructions programed in the NE instruction memory. The NE shared memory serves as the main storage for compressed DNN weights and also the scratchpad memory for input–output activations of layers.

The H264E performs customized H.264 intra-frame compression on an arbitrary (non-rectangular shaped) subset of

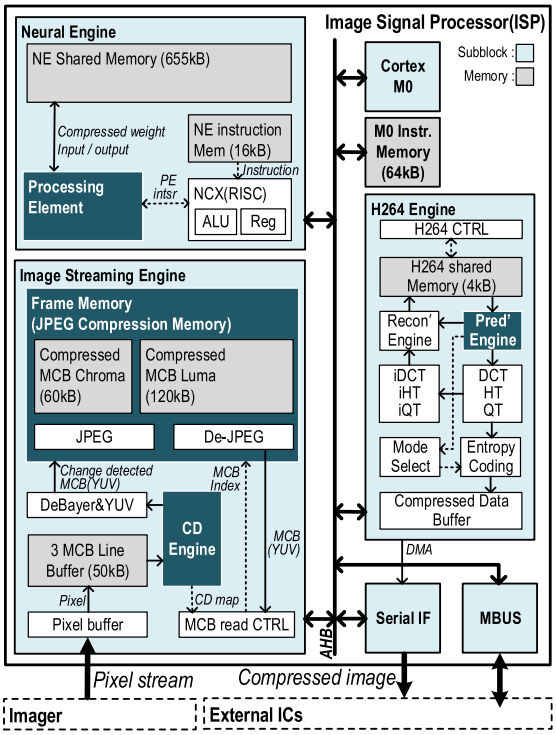


Fig. 2. Architecture of the proposed ISP.

MCBs. The H264E controller automatically collects target MCBs and the boundary pixel information from the ISE block. The prediction and reconstruction engines find the best prediction mode and compress the target MCB with it. The compressed bitstream is transferred off the chip through the serial interface.

An ARM Cortex-M0 orchestrates all the blocks via the AHB bus by executing programs from the M0 instruction memory. The ISP has an MBUS interface [10] for communicating with other chips and the initial programming.

All logic operates in a power-gated 0.6-V domain. The memory banks of the chip have software-controlled separate power-gating switches to optimize system leakage when the ISP is in sleep mode without losing retentive data, such as DNN weights, Cortex-M0, and NE instructions, and reference frame data. The power-gated design enables the ISP to

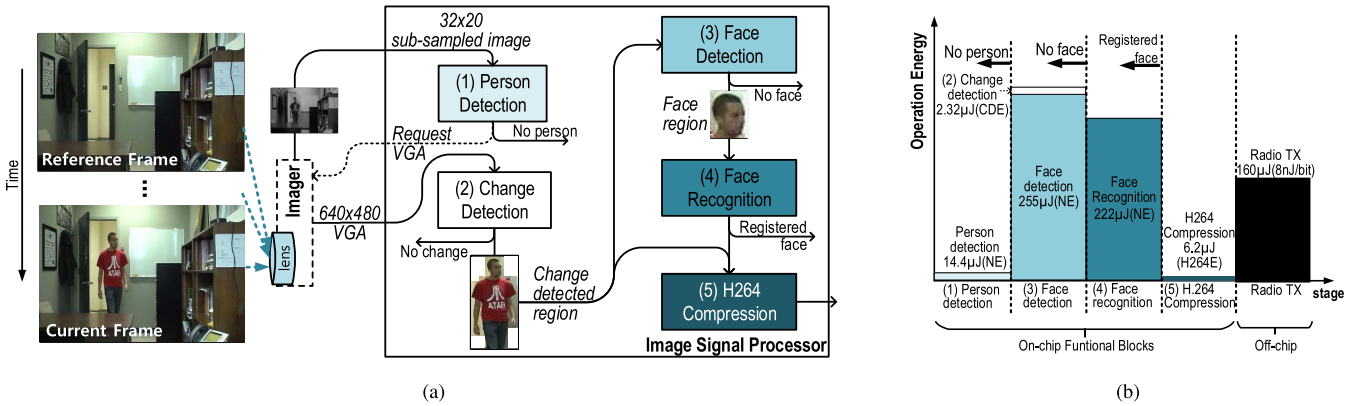


Fig. 3. Use scenario of the proposed ISP enabling hierarchical image recognition. The ISP avoids expending energy on unimportant portions of the incoming frames by screening them out early in the processing pipeline. (a) Example use scenario of the proposed ISP. (b) Operation energy of active functional blocks.

consume only 24 μW while retaining the abovementioned data in the SRAM.

III. Use Scenario

We demonstrate the use of the proposed ISP for intruder detection and recording, as shown in Fig. 3(a). A companion imager chip triggers in response to motion detection [9] to input a sub-sampled image (32 $\times$ 20 pixels $\times$ 1 channel, programmable size empirically chosen for the target ULP person detection accuracy) into the ISP chip. To determine if the sub-sampled image contains a person, the NE performs DNN-based person detection (NE consumes 14.4 μJ). Then, the ISP requests a full Bayer (interleaved RGB) formatted VGA (640 $\times$ 480 pixels) frame from the imager if a person is detected. As the VGA image streams in, the ISP performs on-the-fly MCB-based CD against a previously captured reference frame and compresses the changed blocks using JPEG compression (CDE consumes 2.32 μJ/frame with typical 12% change). Once the ISP has received the whole image, the NE runs DNN-based face detection, which sweeps the region of the changed MCBs on two scales (1$\times$ and 2$\times$ subsampling) with 16-pixel stride (NE consumes 255 μJ). If the NE detects a face (or multiple faces) in the changed region, then the NE runs DNN-based facial recognition (NE consumes 222 μJ) to determine if the face is registered. In the event that the NE does not recognize the face, only change-detected MCBs (not all MCBs) are compressed using H.264 and stored in off-chip flash or radio transmitted. With an average of 12% change-detected MCBs and a 23$\times$ H.264 compression ratio, the ISP achieves 192$\times$ overall size reduction for a VGA frame with 28.3-dB PSNR and only transmits those MCBs with unregistered face information. Once the ISP either finishes transmitting the important portions of the frame or determines that the entire frame is unimportant, it returns to the person-detection step awaiting motion detection trigger at the imager. Fig. 3(b) shows the energy consumption of the active functional blocks for the above-proposed scenario.

By using hierarchical image recognition, we take advantage of the sparsity of new information across an image sequence. The person detection and facial recognition steps allow us to discard unimportant scenes entirely (temporal sparsity), while the CD and face detection isolates the important parts of the image (spatial sparsity). This helps decrease imaging and image transfer energy as the ISP only requests the full image when a person is detected on the subsampled image. Furthermore, this helps reduce flash storage and radio transmission energy consumption.

In addition to the above example scenario, the ISP can adapt to different use-cases and environments by reprogramming to modify the type of information to send off-chip and the specific image recognition DNNs used.

IV. Compression of Memory-Intensive Data

This section introduces the compression techniques of memory-intensive data entities. The ISP requires a large on-chip memory to store frames and algorithm parameters for intelligent image processing. To reduce the SRAM size and, thereby, reduce leakage, we extensively employed data compression. Especially, the input image data, DNN weights, and output image data are all stored or transmitted in compressed format. The combined techniques reduce the on-chip SRAM size by 5$\times$ (from 45 to 9 Mbit) and total leakage power by 9.36$\times$, which includes 2.4$\times$ leakage power reduction via a custom-designed 0.3-V bitcell/0.6-V peripheral SRAM array [11] with 8σ hold margin.

A. Compression of Input Image

On-the-fly JPEG compression is performed on the streamed-in image, achieving an 11.2$\times$ reduction in the required memory size (from 7.4 to 0.66 Mbit) to store two VGA frames (reference and current frames) with 34-dB PSNR. The proposed MCB-based JPEG algorithm and the memory architecture minimize redundant data processing. The 16 $\times$ 16 pixel MCB size is chosen because it is a multiple of JPEG and H.264 unit block sizes (8 $\times$ 8 and 4 $\times$ 4) and also large enough to capture perceptible image features.

The JPEG codec is customized to remove inter-dependence between the MCBs. In Fig. 4(a) illustration, only B0–B3 (8 $\times$ 8 pixels) of an MCB have a dependence, which allows MCB-wise compression and decompression. It provides two

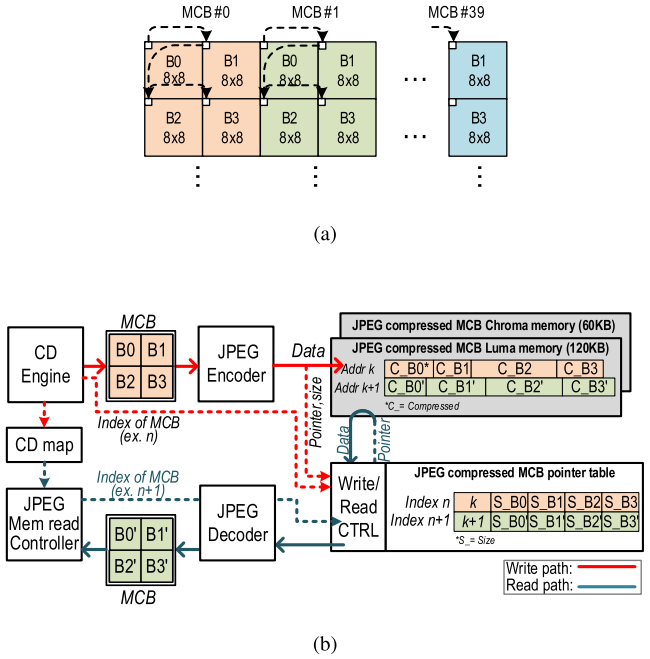
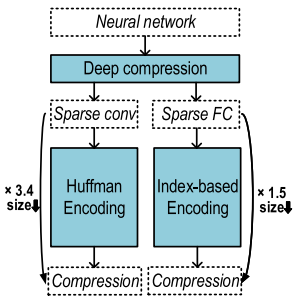


Fig. 4. Proposed compression scheme of input image using single MCB-based JPEG. (a) Customized single MCB-based JPEG. (b) Proposed JPEG compression memory.



Algorithm	Layers	Weight dimension	Density	Compressed parameter (Mbits)
Person detection	conv1	(16,1,3,3)	37.50%	0.007
	conv2	(32,16,3,3)	4.73%	0.017
	fc1	(64, 1280)	2.57%	0.650
	fc2	(2,64)	3.91%	0.008
Face detection	conv1	(16,1,5,5)	40.25%	0.011
	conv2	(32,16,3,3)	4.69%	0.013
	fc1	(64, 2048)	2.30%	0.817
	fc2	(2,64)	5.47%	0.008
Face recognition	conv1	(48,1,5,5)	36.90%	0.025
	conv2	(96,48,3,3)	9.92%	0.113
	conv3	(192,96,3,3)	4.48%	0.363
	conv4	(128,192,3,3)	5.95%	0.498
	conv5	(128,128,3,3)	9.98%	0.363
	fc1	(256,512)	7.46%	0.555
	fc2	(8,256)	9.86%	0.008
Total				3.457

(b)

Fig. 5. Proposed compression scheme of neural-network weights. (a) Weight compression scheme. (b) Compression result of three neural networks.

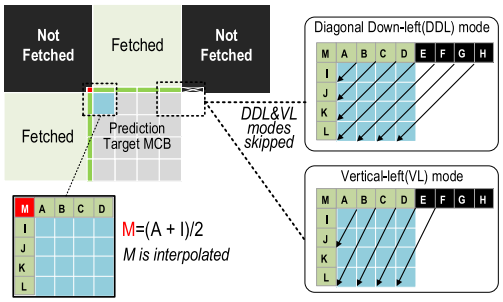


Fig. 6. Proposed H.264 intra-frame compression scheme of output image.

significant benefits. First, since the ISE only compresses change-detected MCBs using JPEG, non-rectangular portions of a changed image can be compressed and stored without redundancy. Second, the other IPs, such as the H264E and NE, can access an arbitrary MCB in the RoI without decompressing the entire frame.

We designed the JPEG compression memory using a pointer-based data structure to accommodate the variable length of the compressed MCBs, as shown in Fig. 4(b). While MCBs are JPEG encoded with a tunable quality factor, the pointer of each MCB contains the starting address of the compressed MCB and the size of four JPEG compression units, C_B0–C_B3. The other IPs can access arbitrary MCBs in raw uncompressed format with natural (fixed-length) block addressing as the decompression happens on the fly. For reading an arbitrary MCB in the current frame, the decompression engine first checks if the MCB to be read is flagged as changed. If so, the MCB of the current frame is decompressed and loaded by referring to the pointer table. Otherwise (the MCB is unchanged), the MCB at the same position of the reference frame is loaded as the proxy of the current frame MCB (not stored in SRAM).

B. Compression of Neural-Network Weights

DNN weights are compressed and stored in on-chip memory. The compressed DNNs are decompressed on the fly when needed by a certain layer. As shown in Fig. 5, we adopted deep compression techniques to optimize the precision and the number of non-zero weights [12]. The value of a non-zero weight and the run length to the next non-zero weight are separately Huffman-encoded. For sparse convolutional layers, each run-length coded non-zero weight requires 13 bits (8-bit weight value and 5-bit run length). The Huffman coding reduces it to 2.47–5.2 bit per non-zero weight. To program sparse convolution weights, a set of convolution kernels are grouped and padded together to form a 512-bit loading unit. This grouping is to increase the area efficiency of the custom ULL SRAM macro and to enable high utilization of the multiplier-and-accumulator (MAC) array. With the density specified in Fig. 5, the overall encoding scheme achieves 3.4× size reduction from 8 to 2.3 bit per weight on average for all convolutions layers in three neural networks. This includes the overhead of SRAM word padding, as well as the storage for Huffman table and tree structure data. As for the sparse fully connected layer, an index-based encoding, described in Section VI-B, is adopted to achieve 1.5× size reduction. With this compression, all three neural networks achieve <1% accuracy degradation compared with uncompressed networks. The compressed weights for all three DNNs used in the intruder detection scenario (680 kbit for person detection, 850 kbit for face detection, and 1.9 Mbit for face recognition) are stored on the chip.

C. Compression of Output Image

The H.264 intra-frame compression algorithm reduces the size of the bits to be transferred off-chip. With an average of 12% change-detected MCBs and 23× H.264 compression ratio, the ISP achieves 192× overall size reduction for a VGA frame with 28.3-dB PSNR. The algorithm is customized for hardware efficiency.

Fig. 6 shows the proposed H.264 intra-frame compression of an output image. The customized H.264 algorithm reduces

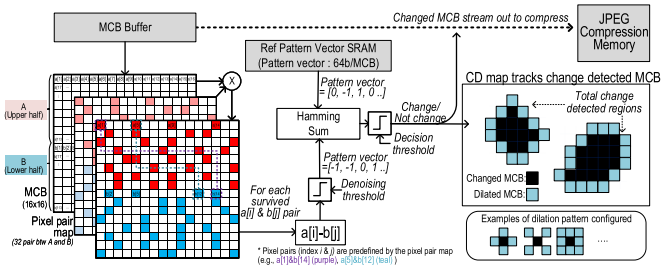


Fig. 7. Proposed MCB-based CD algorithm using sparse census transform encoding. The pixel pair map is a set of pre-configured randomly selected 32-pixel pairs between the top eight rows (A, red) and the bottom eight rows (B, blue) of an MCB. The large dilation pattern improves coverage performance at the cost of increased false detection. The same re-configurable parameters (pixel pair map, denoising threshold, decision threshold, and dilation pattern) are commonly applied to all MCBs.

the number of MCBs required from the JPEG compressed memory for the H.264 intra-mode prediction by interpolating the upper left corner pixel and skipping Diagonal Down Left and Vertical Left prediction modes. This reduces the number of required MCBs by 2.6× with negligible loss (<0.1-dB PSNR) when the changed MCB ratios are 12% of cases. In addition, the customized algorithm enables the arbitrary shape of RoI to be compressed/decompressed.

V. Spatial Image Pruning Using CDE

This section introduces the spatial image pruning using the CDE. We propose a sparse census pattern-based CD algorithm and optimized hardware for processing the algorithm on streamed-in pixels effectively. The combination detects the changed region with low overhead and narrows down the RoI to process.

A. MCB-Based Scene CD Algorithm

The CDE performs a proposed CD at the MCB level of Bayer (interleaved RGB) images for spatial pruning, as shown in Fig. 7. First, the CDE encodes each 16 × 16 pixel MCB (3072 bit) of a reference image into a 64-bit pattern vector. Each element of a pattern vector is the ternary comparison result of two pixels' intensities at predefined positions of the MCB specified as the pixel pair map (same for all MCBs). The tunable denoising value is used for thresholding the comparison result. This new sparse census transform encoding is tolerant of uniform illumination change. For every newly streamed-in image, a 64-bit pattern vector is prepared and compared with that of the reference image. The CDE flags an MCB as changed when the Hamming distance between two vectors exceeds a tunable threshold. To improve coverage, the flagged MCBs are also dilated (neighboring MCBs are flagged in a tunable manner). At the same time, only flagged MCBs are JPEG compressed by the JPEG compression memory block, which is described in Section IV-A. Note that the system can be reprogrammed with a newly captured reference image when the environment changes.

The proposed CD algorithm achieves 95% coverage and a 5% false positive rate on CDnet [13], as shown in Fig. 8.

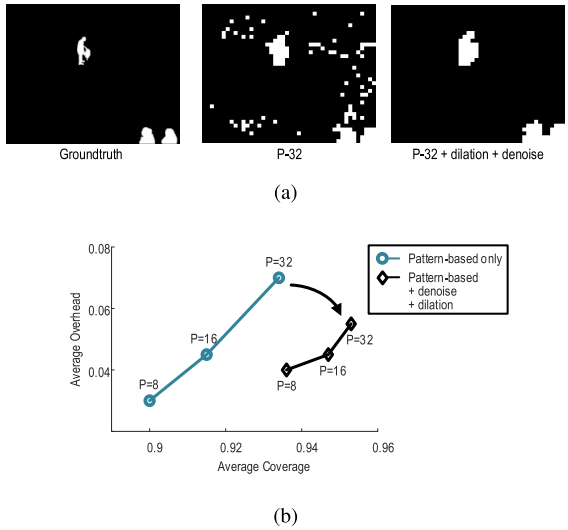


Fig. 8. Evaluation of the proposed CD algorithm. "P" specifies the number of pixel pair in an MCB. P = 32 means that 32 pairs of pixels of an MCB are used for generating pattern vector. CDnet is used for evaluation [13]. (a) Example images of CD algorithms. (b) Performance of CD algorithms.

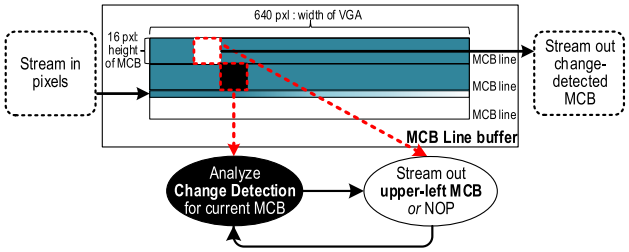


Fig. 9. Proposed buffered MCB-streaming of CDE and the FSM of MCB line buffer.

The denoising and dilation help to reduce overhead and increase coverage from the vanilla pattern-based CD algorithm, especially for a large number of the pattern ($P = 32$). The CD algorithm together with JPEG compression reduces the on-chip VGA image size by 110× from 460 to 4.2 kByte (with typical 12% change).

B. Architecture of CDE

The proposed architecture of the CDE performs CD with an arbitrary dilation pattern on streamed-in images in Fig. 9. First, it minimizes the MCB buffer size (line buffer) by using three MCB-lined circular buffers: the three MCB-lines are necessary for dilation processing of streaming-in MCBs. Second, with the buffered MCB-streaming scheme, we can simplify the logic of the CDE while supporting an arbitrary dilation pattern without increasing the MCB buffer size or clock speed. The image data streams in pixel by pixel and the pixels are accumulated in an MCB line buffer to be read into the MCB-based CD logic. The CD analysis and dilated result of the current MCB are tracked in a CD map. Then, only the top-left MCB of the one being currently analyzed is streamed out depending on the value of the tracked CD map.

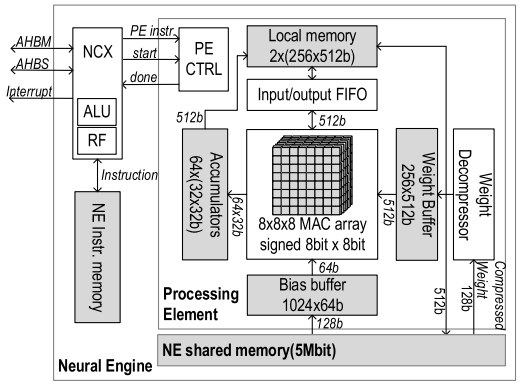


Fig. 10. Architecture of the proposed NE.

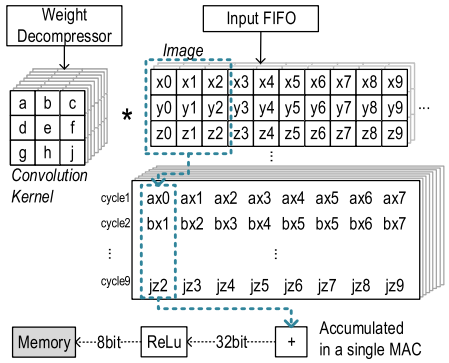


Fig. 11. Computation of the convolution layer.

VI. Event Recognition Using NE

This section introduces event recognition using the NE. The proposed NE accelerates heterogeneous deep neural network operations to enable hierarchical image recognition.

A. Architecture of NE

The highly programmable NE accelerates different types of DNNs. Fig. 10 shows the overall architecture of the NE. The NCX, an NE-dedicated RISC processor, controls the PE by executing instructions from the NE instruction memory. It communicates with other IP blocks via the AHB and interrupts. The PE is a computational core with an 8-bit MAC array and buffers. The weight, bias, and input/output buffers ensure enough data bandwidth for full MAC array utilization. In addition, the NE has 5 Mbit of NE shared memory, a scratchpad of the NCX for storing input/output activation, compressed DNN weights, and bias. The space allocation of the NE shared memory is fully programmable, providing complete flexibility in how the memory space is used. The weight decompressor decodes compressed weights from the NE shared memory and loads them to the weight buffer on the fly.

B. Convolution and Sparse Fully Connected Layers

For a convolution layer, a set of weights is decompressed once and swept across the entire input activation, as shown in Fig. 11. Once the Huffman encoding information of the convolution weights is loaded on the weight decompressor, a set of kernels are decompressed on-the-fly. A MAC array runs MAC operations with decompressed weights and input activations from weight and input/output FIFO each. To save memory for intermediate output activation, the convolved and accumulated ReLU results (32 bit) are shifted back to memory as 8-bit fixed-point values.

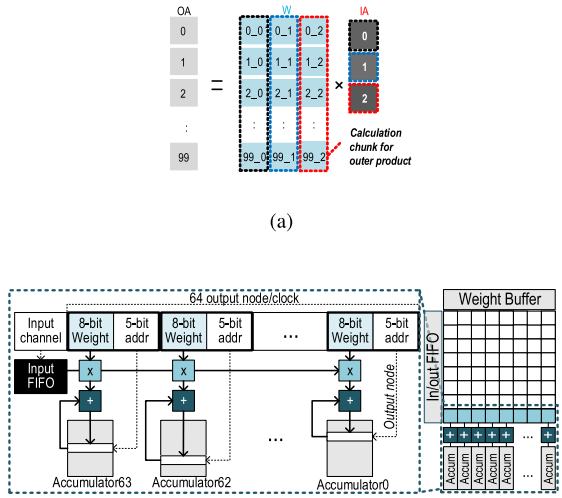


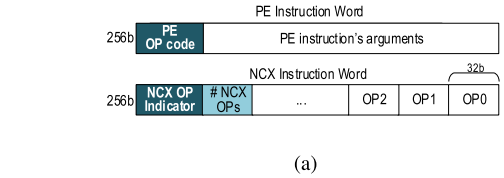
Fig. 12. Operation of sparse fully connected layer. (a) Outer product of matrix-vector multiplication. (b) Computation of sparse fully connected layer.

For a large sparse fully connected layer, as shown in Fig. 12(a), we propose the combination of the outer product of matrix-vector multiplication and index-based weight encoding. In on-demand scenarios for edge intelligent devices, the NE is rarely activated, which lowers the opportunity for batch operations. Each batch typically contains only a single item for the intruder scenario, which disallows the NE from reusing a certain weight more than once, whereas an input activation is used multiple times. Therefore, the outer product-based matrix-vector multiplication is more efficient for sparse fully connected layer operation [14]. In our scheme, only non-zero weights are encoded with their index. The optimized hardware for the outer product maximizes the activity ratio of the MACs by only computing non-zero weights, as shown in Fig. 12(b). In the outer product-based approach, the first input activation element is multiplied to all non-zero elements in the first column of the weight matrix producing a partial sum vector stored in the accumulator memory. The engine proceeds to the next input activation element to be multiplied to all non-zero elements in the second column of the weight matrix. The result is added to the previous partial sum, and this operation continues until all input activations are used. In this way, each element in the input activation vector is maximally reused, unlike the inner product-based approach.

The NE achieves a peak efficiency of 1.5 TOPS/W (two operations = 8-bit multiply and add) at 0.58 V while operating at 153 kHz (allowing 5-frames/s person detection with a 32×20 pixel image).

C. NCX: Custom RISC Unit

The NCX is designed to run NNs independently without complex control from the Cortex-M0 core. The NCX



(a)

NE instruction

Instr	Opcode	Description
PE_CONV	0	Convolution with the given configuration
PE_POOL	1	Max or Avg pool
PE_MOV	2	Move data
PE_ADD	3	Matrix addition of two input matrices
PE_FC	4	Fully-connected layer, sparse(outer product)
PE_DFC	10	Fully-connected layer, dense (inner product)
PE_RELU	5	ReLU on all elements of input
PE_HUFF	6	Load a huffamn table
PE_BIAS	7	Load a set of biases
PE_RESIZE	8	Convolution with a uniform weight
NCX_()	9	Sub-ops to be run on the NCX processor

NCX instruction

Arithmetic				Branch		Special	
Instr	Opcode	Instr	Opcode	Instr	Opcode	Instr	Opcode
ADD	00001	ADDI	00010	BEQ	01001	CP_B	01111
SUB	00011	SUBI	00100	BNE	01010	CP_W	10000
MULT	10001	MULTI	10010	BLT	01011	HALT	11111
MULTS	10011	LSR	11000	BGT	01100	NOOP	00000
AND	10100	LSL	11001	BLE	01101		
OR	10101	LDS	00111	BGE	01110		
XOR	10110	STS	01000				
NAND	10111						

(b)

Fig. 13. NE ISA. (a) Format of NE instruction word. (b) OP codes of NE instructions.

Algorithm 1 Convolution Layer Pseudocode of NCX

```
procedure CONVOLUTION LAYER(ia, w)
    Load Huffman table on weight decompressor
    Load bias on buffer
    while Input activation remains do
        Move input activation to buffer
        Run convolution
        Run ReLU
    end while
end procedure
```

has a dedicated instruction set architecture (ISA), as shown in Fig. 13. The ISA helps to reduce the size of the multiple DNN instruction codes. The 256-bit PE instructions control the PE's operations, such as convolution and the pooling layer. The 32-bit NCX instructions, which are packed in a 256-bit word, do arithmetic, branch, and special operations. Notably, the "copy block (CPB)" instruction loads target image data from the JPEG compression memory to the NE shared memory.

Algorithm 1 presents an example of the convolution layer pseudocode, where a Huffman table and bias are loaded, and then, convolution and ReLU are performed on the loaded input activation.

VII. Other Techniques

A. Reconstruction of H.264 Compressed Bitstream

The H.264 compressed image is reconstructed off-chip from three types of information: the compressed change-detected MCBs, CD map (1200 bits, 1 bit/MCB), and the pre-stored reference image, as depicted in Fig. 14. The CD map indicates the locations of the change detected MCBs. The boundary MCBs of the changed region are decompressed using the reference image. The reference image is compressed and transferred once and then used multiple times until a significant change is detected. The frequency and triggering algorithm for updating

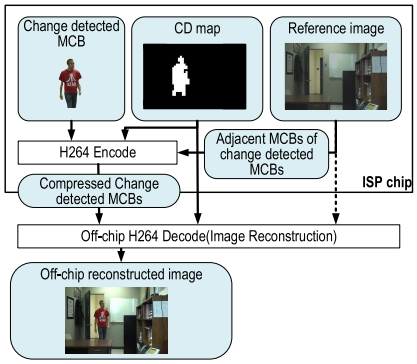
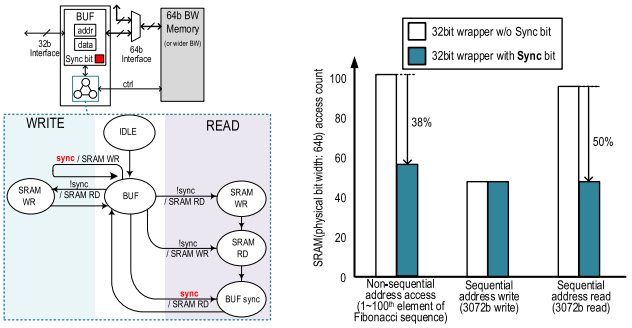


Fig. 14. Proposed image reconstruction scheme of the compressed bitstream.



(a) (b)

Fig. 15. Proposed memory bridge for supporting various memory bandwidth. (a) FSM of memory bridge. (b) Evaluation of bridge with the sync bit.

the reference frame can be programed depending on the use environment.

B. Memory Bridge

We propose a memory bridge with a sync register and a buffer to efficiently access memory banks that provide a larger bandwidth than the bus interface, as shown in Fig. 15. The custom-designed ultra-low leakage SRAM bit-width for a single word was determined to balance the area density of the bit cells, memory access bandwidth, and access energy. As a result, custom-designed SRAM macros in the design have different bit widths of 32, 64, 128, and 256 bits depending on their size and usage. When the SRAM is accessed via a 32-bit-wide bus in the system, it needs a bridge with word buffer registers to resolve the bit-width mismatch. The sync bit associated with the buffer in the bridge indicates whether the current buffered data are synchronized with the address of the accessed memory. When the buffer is synchronized, no additional memory access is needed because accessing the buffer of the bridge is sufficient. For non-sequential 64-bit data memory accesses via a 32-bit bus to execute a Cortex-M0 program, the bridge with a sync bit and buffer reduces the number of memory access by 38% compared with a case without the bridge and buffer. Memory access reduction factor is increased to 50% for sequential memory accesses, which indicates that two consecutive 32-bit bus data are read from a single 64-bit SRAM word. Note that this memory bridge is

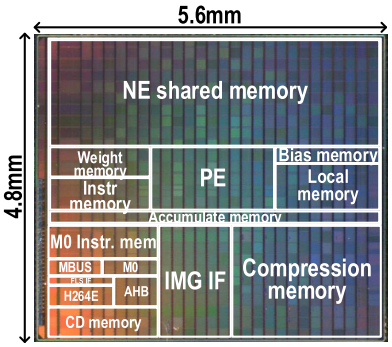
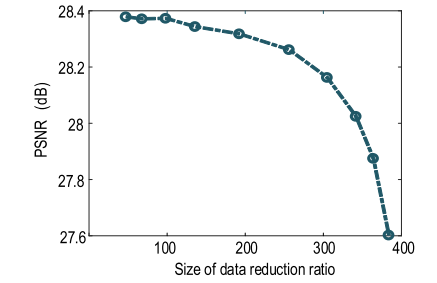
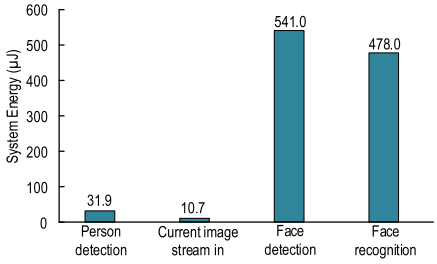


Fig. 16. Die photograph of the ISP.



(a)



(b)

Fig. 17. Measured ISP performance with the condition of 0.58-V logic Vdd and 153 kHz. (CD ratio: 12%.) (a) Output compression performance. (b) System energy consumption.

activated only when the memory is accessed via a central bus. All local memory accesses (for example, within the NE) do not require this memory bridge.

VIII. Measurement Results

The ISP was fabricated in 40-nm LP CMOS, as shown in Fig. 16, and operates at 153 kHz/0.58 V.

The latency (system energy) of person detection, face detection, and face recognition processing is 0.19 s (31.9 μJ), 3.22 s (541 μJ), and 2.85 s (478 μJ), respectively, as shown in Fig. 17. Continually executing each step in the intruder detection and recording scenario [see Fig. 3(a)] consumes 170 μW on average. The energy consumption of the full data flow to produce a 192$\times$ compressed output image (12% MCB change) is 1.5 mJ per frame. The LFW data set [15] and COCO2017 [16] were used for NN training

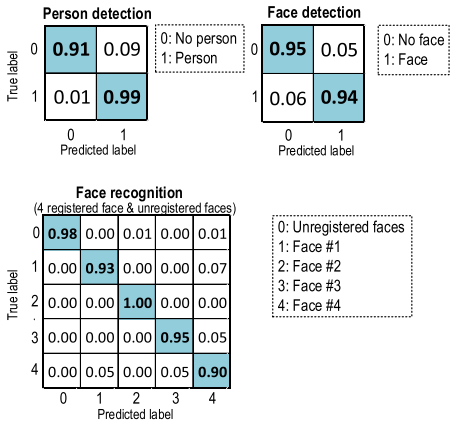
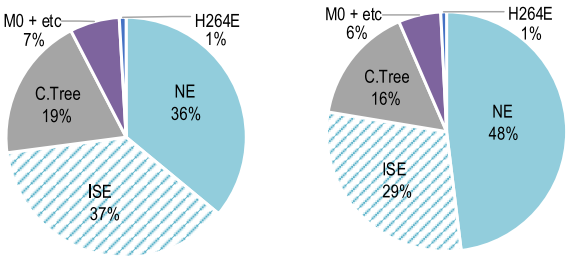


Fig. 18. Confusion matrix of three customized image recognition neural networks.



(a) (b)

Fig. 19. Measured power distribution with the condition of 0.58-V logic Vdd and 153 kHz. (a) VGA-sized current image streaming in with 12% CD ratio (145 μW). (b) Person detection network running (169 μW).

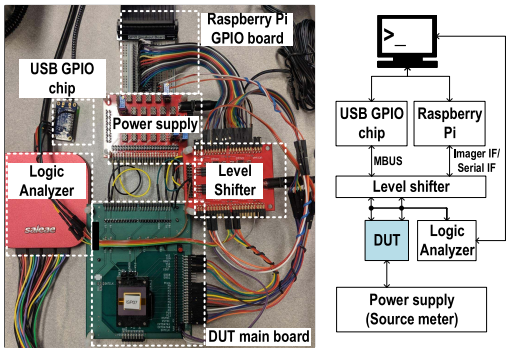


Fig. 20. Testing environment of the ISP.

and testing, yielding the accuracy results given in Fig. 18. When the person detection and unregistered face detection rate are assumed to be 0.005 and 0.0005 frames/s, respectively, the battery-operated system can last 16 times longer than a system that transfers VGA frames wirelessly only when the companion imager detects a triggering motion. Table I shows the comparison to other ASICs that can perform DNN-based face recognition.

The power distributions of the following two main tasks are analyzed: current image streaming and person detection. When the current image is streaming in, the ISE consumes 37% of the total power (145 μW) and performs CD, and preprocess image and JPEG compression, while the NE consumes 36%.

TABLE I
COMPARISON WITH PREVIOUS WORK

	This work	[5]	[6]
Technology	40 nm	65 nm	65 nm
Die area	4.8 mm × 5.6 mm	4 mm × 4 mm	1.784 mm × 1.784 mm
Image recognition algorithm	Change detection NN-based person detection NN-based face detection NN-based face recognition	Haar-like cascade classifier NN-based face recognition	Haar-like cascade classifier Face alignment NN-based face recognition
Image processing algorithm	Debayer, RGB-to-YUV, JPEG, H.264	N/A	N/A
On-chip memory	9 Mbit	1.3 Mbit	0.056 Mbit
NN weight compression	Y (three NNs)	N	N
External memory access	N	N	Y
NN bit precision	8	16	1
Peak energy efficiency	1.5 TOPS/W (NE)	N/A	13.3 TOPS/W
Max resolution	VGA	QVGA	N/A
Power / FPS	170 μW (5 fps PD, 0.28 fps FD, 0.16 fps FR)*	620 μW (1 fps)	200 μW (1 fps)

* Average power to sequentially execute person detection (PD), face detection (FD) and face recognition (FR) at the rate of 5, 0.28 and 0.16 fps respectively.

When the person detection is running, the NE consumes 48% of the total power (169 μW), as shown in Fig. 19.

The ISP was tested in the environment shown in Fig. 20. A Linux machine controls the Raspberry Pi and USB-controlled GPIO board to control the MBUS and imager interface, respectively.

IX. CONCLUSION

We proposed and demonstrated a ULP ISP in 40-nm CMOS technology. The ISP can cut system energy by 16× by transmitting only useful information in a compressed format. Useful information is classified and filtered by enabling hierarchical image recognition in temporal and spatial dimensions. Customized MCB-based compression reduces the size of the image information by 192×. This work demonstrates a complete end-to-end image signal processing platform for mm-scale IoT imaging systems, which includes image preprocessing, recognition, and compression.

REFERENCES

[1] *Ericsson Mobility Report*, Ericsson, Stockholm, Sweden, Jun. 2020.

[2] S. Oh *et al.*, "Iot^2—The Internet of tiny things: Realizing mm-scale sensors through 3 D die stacking," in *Proc. Des., Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 686–691.

[3] Z. Li, J. Wang, D. Sylvester, D. Blaauw, and H. S. Kim, "A 1920×1080 25-frames/s 2.4-TOPS/W low-power 6-D vision processor for unified optical flow and stereo depth with semi-global matching," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1048–1058, Apr. 2019.

[4] J. Oh *et al.*, "Low-power, real-time object-recognition processors for mobile vision systems," *IEEE Micro*, vol. 32, no. 6, pp. 38–50, Nov. 2012.

[5] K. Bong, S. Choi, C. Kim, S. Kang, Y. Kim, and H.-J. Yoo, "14.6 a 0.62 mW ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on haar-like face detector," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 248–249.

[6] S. Kang, J. Lee, C. Kim, and H.-J. Yoo, "B-face: 0.2 MW CNN-based face recognition processor with face alignment for mobile user identification," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 137–138.

[7] S.-L. Chen and E.-D. Ma, "VLSI implementation of an adaptive edge-enhanced color interpolation processor for real-time video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1982–1991, Nov. 2014.

[8] Y. Pu, J. P. de Gyvez, H. Corporaal, and Y. Ha, "An ultra-low-energy multi-standard JPEG co-processor in 65 nm CMOS with sub/near threshold supply voltage," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 668–680, Mar. 2010.

[9] K. D. Choo *et al.*, "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 *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 96–98.

[10] P. Pannuto *et al.*, "MBus: A system integration bus for the modular microscale computing class," *IEEE Micro*, vol. 36, no. 3, pp. 60–70, May 2016.

[11] J. Wang, H. An, Q. Zhang, H. S. Kim, D. Blaauw, and D. Sylvester, "1.03 pW/b ultra-low leakage voltage-stacked SRAM for intelligent edge processors," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2020, pp. 1–2.

[12] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: http://arxiv.org/abs/1510.00149

[13] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.

[14] S. Pal *et al.*, "OuterSPACE: An outer product based sparse matrix multiplication accelerator," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2018, pp. 724–736.

[15] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[16] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48

Hyochan An (Student Member, IEEE) received the B.S. degree in electrical and computer engineering from Sungkyunkwan University, Seoul, South Korea, in 2014. He is currently pursuing the Ph.D. degree with the University of Michigan, Ann Arbor, MI, USA.

From 2014 to 2017, he was a Digital Circuit Engineer with Digital IP Development Team, Samsung Electronics, Hwasung, South Korea. His research interests are energy-efficient deep learning hardware, image signal processors, and neural prosthetic systems.

Mr. An was a recipient of the Doctoral Fellowship from the Kwanjeong Educational Foundation in Korea.

Sam Schiferl is a former graduate student at the University of Michigan, Ann Arbor, MI, USA, under the supervision of Prof. Ron Dreslinski.

He is currently lives and works in Seattle, WA, USA. His research focused on image processing architectures for use in low-power environments.

Siddharth Venkatesan received the B.S.E. degree in computer engineering and the M.S.E. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017 and 2019, respectively.

Currently, he works as a Software Development Engineer at Amazon, Santa Clara, CA, USA, in the Ultra-Fast Grocery Delivery Division. His main interest is in developing algorithms for time-series-based anomaly detection at scale.

Tim Wesley received the B.S. and M.S. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2016 and 2019, respectively.

He is now an Engineer working on AI accelerator chips at a startup.

Qirui Zhang (Graduate Student Member, IEEE) received the B.S. degree (Hons.) from the School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan, Ann Arbor, MI, USA.

His research interests are in low-power and energy-efficient accelerators, processors, and SoCs for mobile robotics and edge intelligence.

Mr. Zhang was a recipient of the World Honorable Mention in the 2016–2017 IEEE Circuits and Systems Society Student Design Competition.

Jingcheng Wang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014, 2017, and 2020, respectively.

He joined Apple, Cupertino, CA, USA, in 2020. He is now working as a Circuit Design Engineer for low-power memory and on-chip sensors.

Kyojin D. Choo (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 2007 and 2009, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2018.

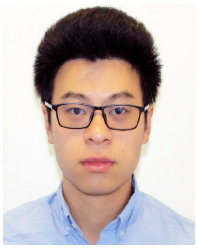
From 2009 to 2013, he was a Circuit Design Engineer with Image Sensor Development Team, Samsung Electronics, Hwasung, South Korea, where he developed signal readout chain for mobile and DSLR image sensors. He is currently a Post-Doctoral Research Fellow with the University of Michigan. His research interests include analog-to-digital converters, energy converter circuits, high-speed communication and timing generation circuits, and millimeter-scale integrated systems.

Shiyu Liu received the B.S.E. degree in computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2020, where she is currently pursuing the master's degree with the Department of Electrical Engineering and Computer Science.

Her research interest lies in computer vision, signal processing, and machine learning.

Bowen Liu (Graduate Student Member, IEEE) received the M.S. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science (EECS).

His research interests include deep learning, computer vision, signal processing, and their applications in low-power systems.



Ziyun Li (Member, IEEE) received the B.S. degree in electrical and computer engineering and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014 and 2019, respectively.

He is currently with Facebook, Redmond, WA, USA. His research interests include high-performance, energy-efficient computer vision/machine learning processing units to enable next-generation intelligent and autonomous vision systems for AR/VR.

Dr. Li was a recipient of the Best Paper Award at the 2016 IEEE Workshop on Signal Processing Systems.

Luyao Gong received the B.S. degree in electrical engineering and computer science from Tianjin University, Tianjin, China, in 2017, and the M.S. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2018.

Currently, she works for Google, Mountain View, CA, USA.

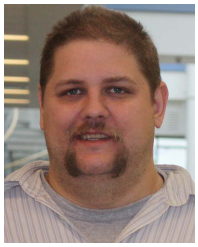
Hengfei Zhong received the B.Eng. degree in microelectronics from Sun Yat-sen University, Guangzhou, China, in 2016, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018.

He is currently with Silicon Engineering Group, Apple Inc., Cupertino, CA, USA. His work focuses on CPU design and verification, across multiple units, such as CPU core, cache, and memory subsystems.

David Blaauw (Fellow, IEEE) received the B.S. degree in physics and computer science from Duke University, Durham, NC, USA, in 1986, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1991.

Until August 2001, he worked for Motorola, Inc., Austin, TX, USA, where he was the manager of the High Performance Design Technology Group and won the Motorola Innovation Award. Since August 2001, he has been on the faculty of the University of Michigan, Ann Arbor, MI, USA, where he is the Kensall D. Wise Collegiate Professor of electrical engineering and computer science. He is the Director of the Michigan Integrated Circuits Lab. He has published over 600 articles and has received numerous best paper awards and he holds 65 patents. He has researched ultralow-power wireless sensors using subthreshold operation and low-power analog circuit techniques for millimeter systems. This research was awarded the MIT Technology Review's "one of the year's most significant innovations." His research group introduced so-called near-threshold computing, which has become a common concept in semiconductor design. Most recently, he has pursued research in cognitive computing using analog, in-memory neural networks for edge-devices and genomics for precision health.

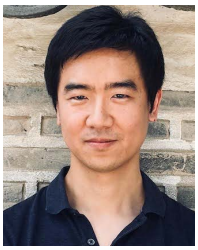
Dr. Blaauw received the 2016 SIA-SRC Faculty Award for lifetime research contributions to the U.S. semiconductor industry. He was the General Chair of the IEEE International Symposium on Low Power and a member of the IEEE International Solid-State Circuits Conference (ISSCC) Analog Program Subcommittee.



Ronald Dreslinski (Senior Member, IEEE) received the B.S.E., M.S.E., and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2001, 2003, and 2011, respectively.

He is the Morris Wellman Faculty Development Assistant Professor of computer science and engineering at the University of Michigan. His work focuses on hardware and circuit designs for a post-Moore's Law world.

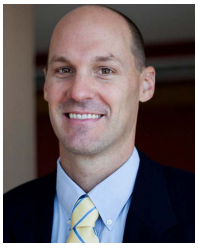
Dr. Dreslinski received the 2015 IEEE Young Computer Architect Award.



Hun Seok Kim (Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 2010.

He is currently an Assistant Professor with the University of Michigan, Ann Arbor, MI, USA. His research focuses on system analysis, novel algorithms, and very-large-scale integration (VLSI) architectures for low-power/high-performance wireless communications, signal processing, computer vision, and machine learning systems.

Dr. Kim was a recipient of the 2018 Defense Advanced Research Projects Agency (DARPA) Young Faculty Award (YFA) and the National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award 2019. He is an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and the IEEE SOLID STATE CIRCUITS LETTERS.



Dennis Sylvester (Fellow, IEEE) received the Ph.D. degree in electrical engineering from UC-Berkeley, Berkeley, CA, USA, in 1999.

He is the Edward S. Davidson Collegiate Professor of electrical and computer engineering at the University of Michigan, Ann Arbor, MI, USA. His main research interests are in the design of miniaturized ultralow-power microsystems, touching on analog, mixed-signal, and digital circuits. He has published over 500 articles and holds more than 50 U.S. patents in these areas. His research has been commercialized via three major venture capital-funded startup companies: Ambiq Micro, Cubeworks, and Mythic.

Dr. Sylvester has received 11 best paper awards and nominations and was named a Top Contributing Author at ISSCC and the most prolific author at the IEEE Symposium on VLSI Circuits. He is currently a member of the Administrative Committee of the IEEE Solid-State Circuits Society, an Associate Editor of the IEEE JOURNAL OF SOLID-STATE CIRCUITS, and was an IEEE Solid-State Circuits Society Distinguished Lecturer for 2016–2017. He held research staff positions at Synopsys and Hewlett-Packard Laboratories as well as visiting professorships at the National University of Singapore and Nanyang Technological University.