24-GOPS 4.5-mm^2 Digital Cellular Neural Network for Rapid Visual Attention in an Object-Recognition SoC

Seungjin Lee, *Student Member, IEEE*, Minsu Kim, *Student Member, IEEE*, Kwanho Kim, *Member, IEEE*, Joo-Young Kim, *Member, IEEE*, and Hoi-Jun Yoo, *Fellow, IEEE*

***Abstract*— This paper presents the Visual Attention Engine (VAE), which is a digital cellular neural network (CNN) that executes the VA algorithm to speed up object-recognition. The proposed time-multiplexed processing element (TMPE) CNN topology achieves high performance and small area by integrating 4800 (80 × 60) cells and 120 PEs. Pipelined operation of the PEs and single-cycle global shift capability of the cells result in a high PE utilization ratio of 93%. The cells are implemented by 6T static random access memory-based register files and dynamic shift registers to enable a small area of 4.5 mm^2. The bus connections between PEs and cells are optimized to minimize power consumption. The VAE is integrated within an object-recognition system-on-chip (SoC) fabricated in the 0.13-μm complementary metal–oxide–semiconductor process. It achieves 24 GOPS peak performance and 22 GOPS sustained performance at 200 MHz enabling one CNN iteration on an 80 × 60 pixel image to be completed in just 4.3 μs. With VA enabled using the VAE, the workload of the object-recognition SoC is significantly reduced, resulting in 83% higher frame rate while consuming 45% less energy per frame without degradation of recognition accuracy.**

***Index Terms*— Cellular neural network, object-recognition, saliency map, visual attention.**

I. Introduction

RECENTLY, there has been increasing interest in vision chips for pattern recognition applications such as autonomous vehicles, mobile robots, and human–computer interfaces [1], [2]. These chips must be able to execute complex vision algorithms in real time while consuming as little power as possible for long battery life. This is difficult to achieve due to the high computational cost of vision algorithms. For example, a popular local-feature-based object-recognition algorithm [3] requires nearly 1 s to process a single 640 × 480 pixel frame on a modern personal computer. The prevalent trend for performance enhancement has been to simply increase the level of parallelism, many of the recently proposed systems-on-chip (SoCs) contain tens to hundreds of processors operating in the multiple-instruction multiple-data mode [1] to exploit task-level parallelism, or the single-instruction multiple-data (SIMD) mode [2] to exploit data-level parallelism. However, regardless of the configuration, the increase in processor count comes at the cost of increased power consumption, which not only drains precious battery power but also increases the heat dissipation requirements of the system. Thus, a new approach that can speed up vision algorithms while simultaneously reducing power consumption is highly desirable.

In this paper, we present the visual attention engine (VAE) [4], which is a hardware accelerator optimized for the saliency-based VA algorithm [5], which mimics the VA mechanism of the human brain [6]. The object-recognition flow with VAE is illustrated in Fig. 1. The VAE executes the saliency-based VA algorithm prior to the detailed feature extraction and feature matching stages, thereby reducing their workload. As a result, the VAE not only reduces the processing time required per frame of object-recognition but also reduces the energy required per frame.

The VAE is implemented as part of a multiprocessor object-recognition SoC [7] that integrates the VAE together with eight processing element clusters (PECs), a matching accelerator (MA), and a host RISC processor as shown in Fig. 2 (chip photograph shown in Fig. 15). The VAE executes the saliency-based VA algorithm on the input image to obtain regions of interest (ROIs) for further processing. The eight PECs, which are each responsible for one-eighth of the input image, perform scale-invariant feature transform (SIFT) [3] feature extraction on the ROIs in parallel. The MA performs nearest neighbor matching of the extracted SIFT features against an object database. The RISC processor performs task management and flow control.

Since the VAE is an added stage to the conventional pattern recognition flow, its processing speed must be sufficiently high in order to minimize the latency of the VA algorithm, which itself requires a considerable amount of computations. Also, it should be energy efficient and occupy a small area since it is targeted for integration within an object-recognition SoC. Cellular neural networks (CNNs) [8] are known to be a very

Manuscript received October 13, 2009; revised September 18, 2010; accepted September 20, 2010. Date of publication November 11, 2010; date of current version January 4, 2011.

S. Lee, M. Kim, J.-Y. Kim, and H.-J. Yoo are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: seungjin@eeinfo.kaist.ac.kr; beatin@eeinfo.kaist.ac.kr; trample7@eeinfo.kaist.ac.kr; hjyoo@ee.kaist.ac.kr).

K. Kim was with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea. He is currently with the Digital Media & Communication Research and Development Center, Samsung Electronics, Gyeonggi-do 443-742, Korea (e-mail: kimkh82@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2085443

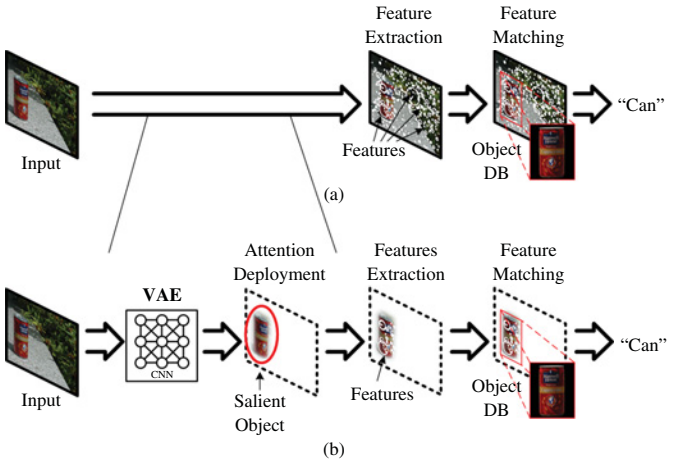


Fig. 1. Steps of object-recognition (a) without VA and (b) with VA. VA can speed up object-recognition by reducing the number of features that must be analyzed by subsequent steps.

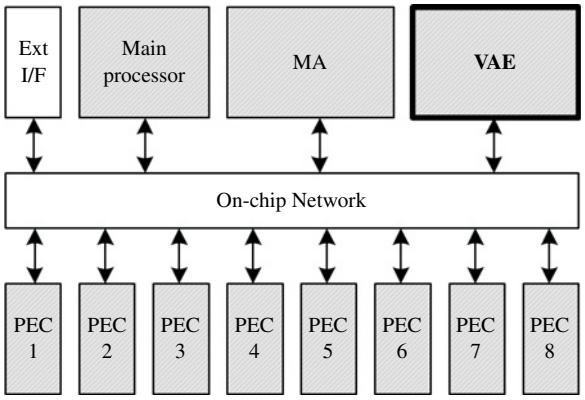


Fig. 2. Object-recognition SoC including the VAE. The VAE performs saliency-based VA to reduce the overall workload of the entire SoC.

efficient hardware architecture for executing the saliency-based VA algorithm. Their locally connected topology is optimally suited for local convolution operations such as Gaussian and Gabor filtering. Thanks to CNN's propagation property, even operations requiring large kernel sizes are readily supported, as demonstrated in [9], in which Gabor-like filtering that requires very large kernel size is achieved on CNNs using just 3×3 sized templates.

For the VAE, we chose a digital CNN implementation that is easily integrated within the multiprocessor SoC. The VAE implements a new CNN architecture integrating 120 PEs interleaved with 4800 (80×60) cells. Named the time-multiplexed processing element (TMPE) topology, it allows the VAE to achieve very high PE utilization and thus high performance. As a result, the VAE is able to execute the VA algorithm in a short time, making the execution time overhead of calculating VA small compared to the large reductions in execution time of the main object-recognition. This paper is organized as follows. Section II will introduce the saliency-based VA algorithm, which is the basis for this paper. Section III will describe the architecture of the VAE, starting with the derivation of the TMPE CNN topology. The detailed implementation of the VAE circuits will be discussed in Section IV, followed by the detailed operation of the VAE in Section V. Section VI will give the implementation details with chip measurements.

II. VA Algorithm

In the human brain, VA [6] is the mechanism responsible for focusing the limited resources of the brain on the most important information in the visual field. The same principle can be applied to machine vision, as it is very challenging to perform object-recognition on real-time video inputs using complex algorithms such as SIFT, especially when the field of view is cluttered with distractors.

In order for VA to be useful for object-recognition, it must be able to select target objects while rejecting regions containing distractors. This paper employs the saliency-based VA model proposed by Itti, Koch, and Niebur [5]. On top of the previous work of Koch and Ullman [10], the model is based on the observation that attention in primate vision is asserted as a result of competition between salient features in the visual stimuli. Testing on natural images has shown that the saliency-based model closely mimics human VA behavior [11]. In addition, its practical usefulness to object-recognition is demonstrated in [12], where it is used to pick out target objects superimposed onto natural backgrounds.

The detailed steps of the VA algorithm are illustrated in Fig. 3. The resolution of the input image of the algorithm is 80×60 pixels, or one-fourth of the 320×240 pixel image used for object-recognition. The one-fourth scaled image of 80×60 pixels corresponds to the highest resolution image required by the saliency-based VA model [5]. The most time-consuming calculations of the VA algorithm are the Gabor filtering, image wide normalization, and Gaussian filtering. These are the main operations targeted by the VAE.

III. Architecture

A. CNNs

CNNs are a type of neural network composed of a 2-D cell array with each cell having synapse connections to and from all cells in its neighborhood [8]. The following is the differential equation describing the operation of a CNN cell:

$$\frac{dx^c(t)}{dt} = -x^c(t) + \sum_{d \in N_r} a_d y^d(t) + \sum_{d \in N_r} b_d u^d + i \quad (1)$$

$$y^c(t) = \begin{cases} -1, & x^c(t) < -1 \\ x^c(t), & -1 \le x^c(t) \le 1 \\ 1, & x^c(t) > 1. \end{cases} \quad (2)$$

The variable x^c is the internal state of the cell, y^c is its output, and u^c is its input. N_r denotes the neighborhood of the cell, which is usually the 3×3 window (including itself) centered around the cell. Thus, x^d, y^d, and u^d denote the internal state, output, and input, respectively, of neighborhood cells. The coefficients a_d, b_d, and i are constants that make up a template which defines the behavior of the CNN.

Equations (1) and (2) describe the operation of CNNs in continuous time and can be directly modeled by analog

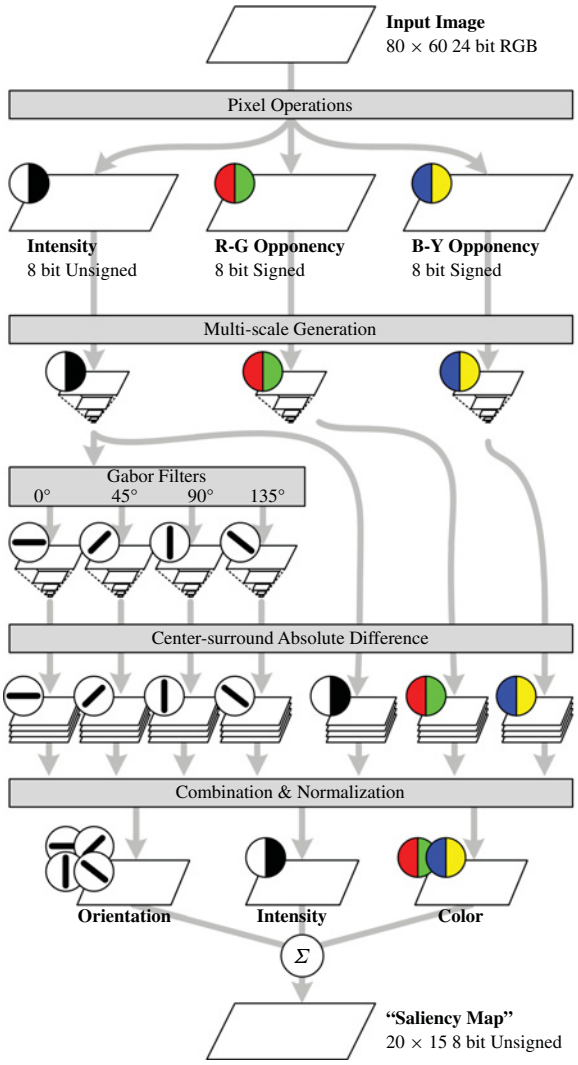


Fig. 3. Detailed steps of the saliency-based VA algorithm. The final resulting saliency map indicates the regions of the input image that are most likely to be important.

circuit components. In digital CNNs, the first-order Euler approximation of (1) and (2) is used, given as

$$x^c(k+1) = \sum_{d\in N_r} a_d y^d(k) + \sum_{d\in N_r} b_d u^d + i \tag{3}$$

$$y^c(k) = \begin{cases} -1, & x^c(k) < -1 \\ x^c(k), & -1 \le x^c(k) \le 1 \\ 1, & x^c(k) > 1. \end{cases} \tag{4}$$

By examining (1)–(4), it can be seen that CNN hardware must provide memory, inter-cell communication, and processing capabilities. Memory is needed to store the state and input of the cells, inter-cell communication is needed to access the states and inputs of neighbor cells, and processing is needed to apply CNN templates. Fig. 4(a) and (b) summarizes the topologies of conventional analog and digital CNN implementations. Analog CNNs employ a fully parallel architecture in which each cell has its own set of analog multipliers and memory [13], [14]. This results in very fast operation on the order of several hundred 8-bit equivalent GOPS [13]. However,

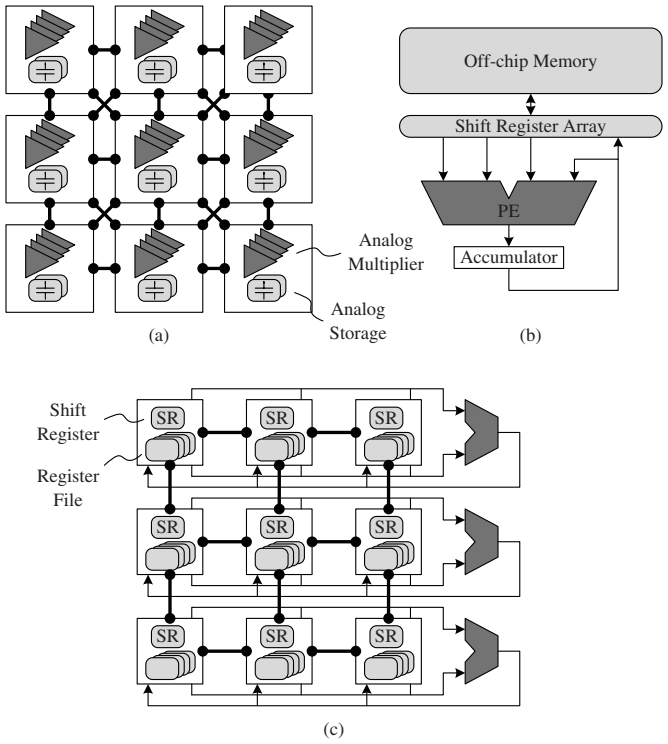


Fig. 4. Comparison of CNN implementations. (a) Conventional analog. (b) Conventional digital. (c) TMPE topology.

actual performance is limited by the I/O speed, which is only in the several hundred MByte range, due to digital-to-analog converter and analog to digital converter overhead. In addition, analog CNNs suffer from switching noise and low equivalent accuracy in the range of 7 b.

Meanwhile, digital CNNs replace analog multipliers with digital multiplier-accumulators (MACs) units, and analog memories with static random access memory (SRAM) and flip-flops for more robust operation. However, due to the large size of MAC units compared to analog multipliers, it is not possible to employ a fully parallel architecture. In [15], a systolic array architecture was proposed to emulate CNN operation. It provides only limited scalability due to its complex architecture and its performance is bottlenecked by its dependence on off-chip memory accesses. More recently, a mixed-mode approach was presented [16], in which inter-cell connections and accumulation are handled in the analog domain while memory storage is handled in the digital domain. Although this approach seems to take the best of analog and digital circuits, its actual performance suffers from the long digital-to-analog and analog to digital conversion time. In [17], a processor array approach is used in which each processor is responsible for a portion of the image. The processors are controlled in a SIMD way, meaning a single controller is responsible for decoding program instructions and broadcasting them to each processor.

We propose a digital CNN topology, named the TMPE topology, which combines the flexibility of the digital CNN approach with the high performance of a fully parallel cell topology adopted by analog CNNs. This is achieved by integrating the required number of fully parallel cells with

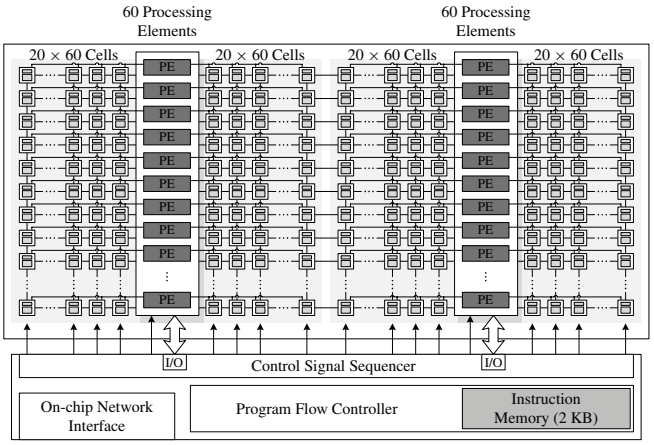


Fig. 5. Block diagram of the VAE. A total of 120 PEs are shared by 4800 cells.

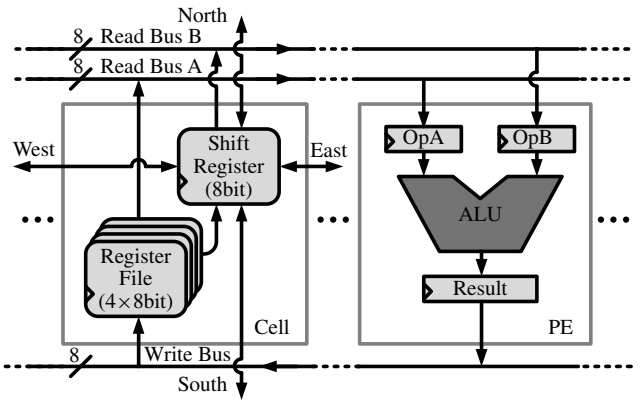


Fig. 6. Cell-PE interconnection. Each PE can access 40 cells through two read buses and one write bus.

a smaller number of shared digital PEs, as illustrated in Fig. 4(c). In this highly scalable topology, the cell array size can easily be scaled because of the small size of each cell. Processing speed can be scaled as needed by varying the number of PEs per row of cells. Meanwhile, the parallel inter-cellular communication capability of the cells minimizes communication overhead to ensure that the PEs are always optimally utilized. The detailed operation of this topology will be further explained in Section V.

B. VAE Architecture

Fig. 5 shows the block diagram of the VAE which implements the TMPE CNN topology. The 4800 (80×60) cells required to execute the VA algorithm are organized into four 20×60 cell arrays, while the 120 PEs that provide a peak performance of 24 GOPS at 200 MHz are organized into two columns of 60 PEs. Each column of PEs is shared by two cell arrays, which results in each PE being shared by 40 cells.

The simplified block diagram of the cells and PEs are shown in Fig. 6. A strict role division between the cells and PEs was enforced, the cells are responsible for the storage and inter-cellular communication of CNN variables, while the PEs handle the calculations. This means that the PEs do not even have an accumulation register which is required for CNN operation. Instead, accumulation is performed directly on each cell's register file. The lack of an accumulation register within the PE is due to the time-multiplexed sharing of each PE among different cell columns.

Each cell contains a register file with four 8-bit entries for storing the variables required for CNN operation. Although (3) and (4) require each cell to store three variables, i.e., u^c, x^c, and y^c, only two variables are needed when applying the full signal range (FSR) model [18], in which $y^c = x^c$. As a result, the four entry register files can store up to two real-valued CNNs. Complex-valued CNN operation, which is required for Gabor-type filtering [9], requires all four register file entries, two for storing the real and imaginary parts of the state $x^c (x^c = x^c_{\text{real}} + j x^c_{\text{imag}})$, one for storing u^c, and one as a temporary register when calculating the complex convolution. The quad-directional shift registers of each cell are crucial to the VAEs operation since they are the only means of inter-cell communication. For CNN operation, either the input u^c or state x^c is loaded from the register file into the shift register and shifted to neighboring cells in the north, east, west, and south directions. Since all shift registers operate in unison, as a cell's data is shifted out to a neighbor in a certain direction, data from the neighbor in the opposite direction will be shifted in. Each global shift operation takes only one cycle to complete, resulting in a very low data access overhead of only 2.4% for CNN operation.

The PEs are responsible for updating the CNN states stored in the cells. For this, two read buses and one write bus connect each PE to 40 cells. When processing data, a PE reads data from the register file and shift register of a cell through read bus A and read bus B, and writes back the results to the register file through the write bus. It should be noted that for each read–execute–write operation, the PE has access to only one cell. This means it cannot by itself combine data stored in different cells, that is achieved by the shift registers in the cells. Each PE is capable of executing an 8-b MUL or MAC with result saturation in a single cycle to accelerate CNN operation. This amounts to 24 GMACS peak performance at 200 MHz operating frequency. The sustained performance for CNN operation is only slightly lower at 22.3 GMACS, thanks to high PE utilization of 93%.

The operation of the cells and PEs is coordinated by the control block shown at the bottom of Fig. 5. The VAE program is stored in 2 Kbytes of instruction memory. The program flow controller decodes the instructions and provides basic looping capability. The control signal sequencer generates the low-level timing signals for controlling the cells and PEs. Data I/O of the cells' data takes place through the on-chip network interface.

IV. Detailed Implementation

A. VAE Cell

Area reduction is the most critical aspect of the VAE cell's design, due to the large number of cells that must be integrated. The 4800 cells account for more than 60% of the total area of the VAE. The register file and shift register of the cell are optimized to reduce area. First, the register file is implemented

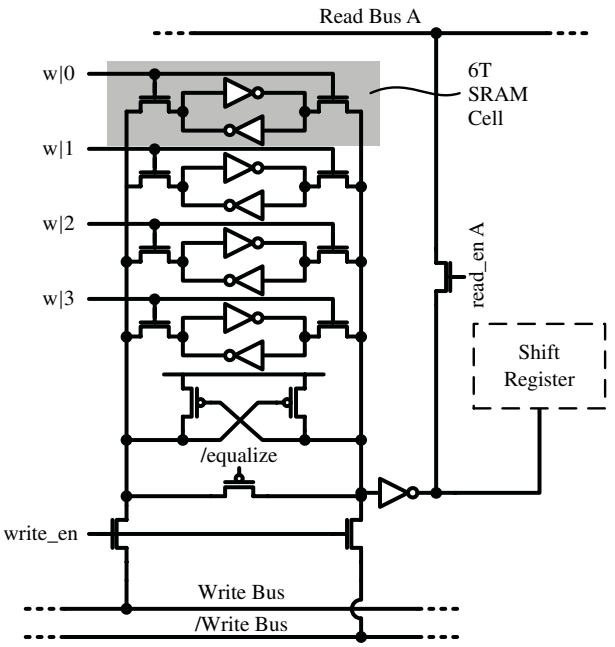


Fig. 7. Bit slice of a VAE cell's register file. 6T SRAM-based storage elements are used for small area.

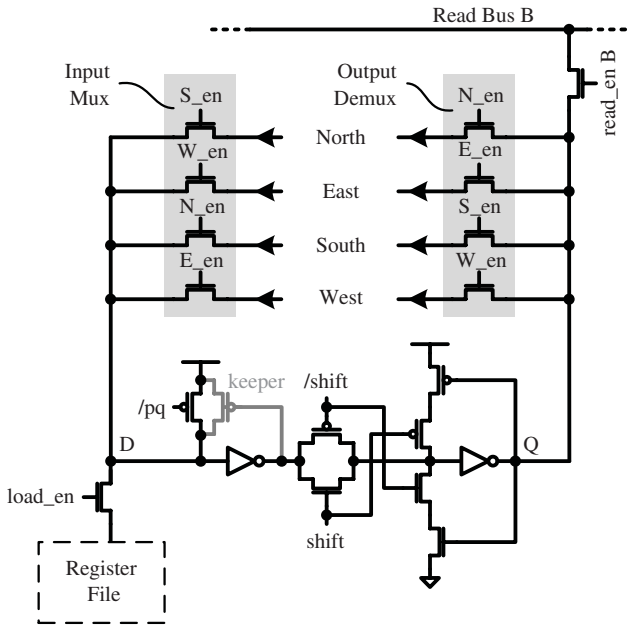


Fig. 8. Bit slice of a VAE cell's shift register. NMOS pass-transistor-based I/O switching is used for small area.

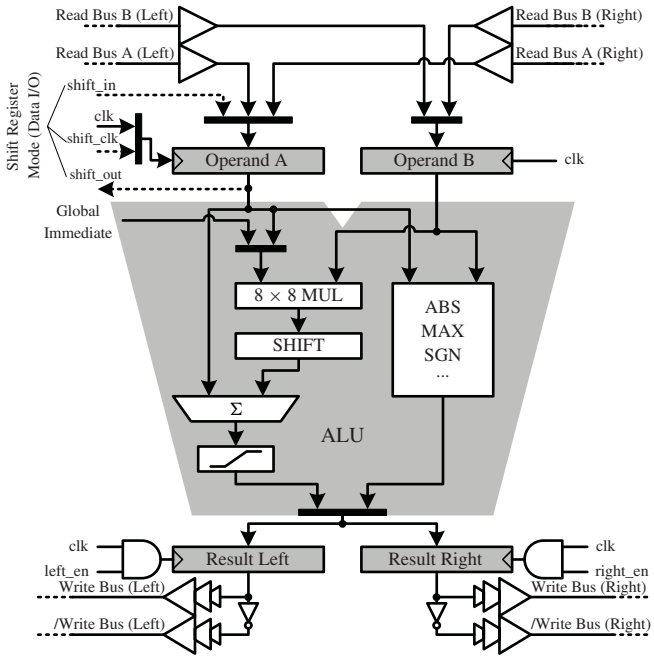


Fig. 9. PE circuit and read/write buses.

using 6T SRAM cells as shown in Fig. 7. To further reduce the area, peripheral circuits such as the asynchronous timing signal generator is not included in each cell. Instead, a global asynchronous timing generator that is located in each 20×60 cell array distributes the asynchronous timing signals to each cell through a balanced H-tree buffer configuration. Power consumption of the buffer tree is minimized by gating the control signals by column, since only two columns of cells are active at once due to pipelined operation. Compared to a standard cell D flip-flop based implementation, 55% cell area reduction is achieved by using SRAM cell-based register files with global timing signal generators.

Cell area is further reduced by employing a dynamic negative-channel metal-oxide-semiconductor (NMOS) pass-transistor based MUX/DEMUX scheme shown in Fig. 8. In this scheme, the value of dynamic node D, which is precharged to the supply voltage, is evaluated through one of many possible paths selected by the signals "N_en," "E_en," "S_en," "W_en," and "load_en" before being latched by the pulsed latch. A weak keeper is employed to protect node D against noise components due to crosstalk and charge sharing. Compared to a static standard cell MUX/DEMUX design that requires positive-channel metal-oxide-semiconductor and NMOS transistors, this reduces cell area by 40%.

B. PE

The PEs execute the basic functions required for CNN operation such as MAC, MUL, and ADDI in a single cycle as shown in Fig. 9. The limiter located after the adder can simulate the nonlinear output function of CNNs. Additional functions such as ABS and MIN/MAX allow for some basic general purpose image manipulations.

Each PE is shared by a group of 40 cells through two read buses and one write bus. The read buses, which are single-ended to save routing resources, are operated in dynamic mode. Since NMOS pass transistors are used in place of complementary metal–oxide–semiconductor (CMOS) pass gates, cell area and bus loading can be reduced. On the other hand, the write buses are double-ended to facilitate reliable write operation to the SRAM cells of the register files.

The read and write buses are split into left and right segments at the PE to reduce wire capacitance and resistance. Originally, each bus wire would have to span a width of 1750 μm, which includes the width of 40 cells and 1 PE. However, with bus splitting, this is reduced to just 570 μm,

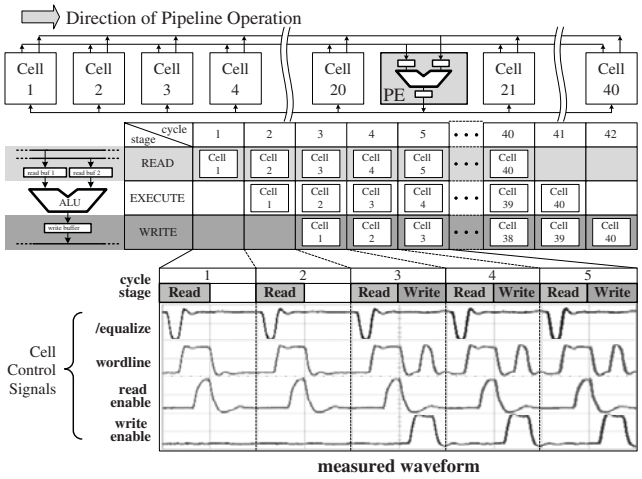


Fig. 10. Pipelined PE operation. Consecutive read and write stages in a single cycle enable 1 op/cycle throughput of the PEs.

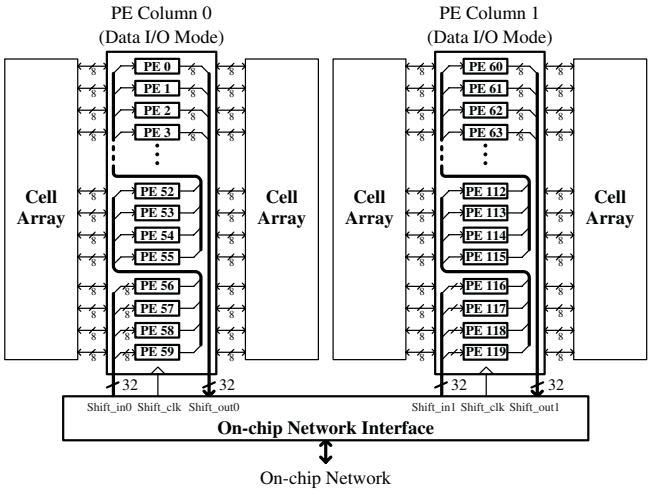


Fig. 11. Data I/O path of the VAE. All data access to the cell arrays are performed through the PEs.

which is the width of just 20 cells. As a result, the bus capacitance, which includes the wire capacitance as well as the parasitic capacitance of the access transistors, is reduced by over 50% and the read bus wire delay from the farthest cell on the bus to the PE is just 240 ps.

Fig. 10 illustrates the pipelined execution of the PEs. The data in the cells are processed column by column in a three-stage read, execute, write pipeline. As a result, 1 op/cycle throughput is achieved and it takes 42 cycles for the PEs to execute one instruction on the entire cell array. Pipelined operation necessitates read and write of register files in the same cycle. However, since control signal generation circuitry is shared among cells in an array, reading and writing of cell data cannot occur simultaneously. This is solved by allocating the first half of each cycle for reading and the second half for writing.

As explained earlier, the PE does not contain an accumulating register but instead accumulates directly to a cell's register file. The downside of this is that only 8 b can be used for accumulation. This poses two potential problems: accumulator overflow and degradation of precision. The overflow problem is solved by scaling the multiplication result with a barrel shifter as shown in Fig. 9. Despite this, the degradation of precision may still limit the VAE in some applications that, for example, require large kernel sizes. However, for the VA application, 8 bits proved to be sufficient.

C. Data I/O

For data I/O, the PEs are connected to one another in a shift register configuration to shift data to and from the controller as shown in Fig. 11. For this, the "Operand A" registers of each PE can be operated in a shift register mode through the shift_in, shift_out, and shift_clk signals, as described in Fig. 9. The "Operand A" registers of four consecutive PEs are grouped as one 4-byte shift register to match the 4-byte per cycle I/O throughput of the controller. For writing to the cell array, an entire column of data is first shifted in to a PE column and then written to a column of cells at once through the write bus. When reading from the cell array, data from an entire column of cells are read into a PE column either through read bus A or read bus B, and then shifted out to the controller. Each column operation takes 1 cycle for cell access and 15 cycles for shifting the data, which translates to 1280 cycles or 6.4 μs for array-wide write or read.

D. Controller

The controller is responsible for decoding VAE programs into control signals for the cell and PE arrays and data input/output. Each VAE instruction is 16 bits wide so that the 2 Kbyte instruction memory can hold 1024 instructions. The VAE instructions are fine grained, meaning that every operation such as load, shift, and ALU operations are programmable. Although this results in larger program size, since CNN templates must be broken down into multiple instructions, it affords high flexibility to the programmer for non-CNN tasks. The task of decoding instructions is split between the flow control block and the sequencer. The flow control block takes care of instruction-level flow control including branching and program start/halt. The sequencer decodes the current instruction into actual sequences of control signals for the cell and PE arrays. The sequence of control signals can be as long as 42 cycles for instructions involving PE processing like multiply-accumulate or just 1 cycle for instructions like load and shift.

V. Detailed Operation

Fig. 12 shows the programmer's model of the VAE. From the programmer's viewpoint, the VAE can be abstracted into a vector machine with four 80 $\times$ 60-dimensional 8b vector registers, denoted r0 $\sim$ 3, and one 80 $\times$ 60-dimensional 8b vector shift register, denoted sr. Three main types of operations are possible on the VAE. First, the vector shift register contents can be loaded from one of the vector registers. This takes place globally for all cells and thus only one cycle is needed. Second, the contents of the vector shift register can be shifted in the north, east, south, or west directions. This also takes just one

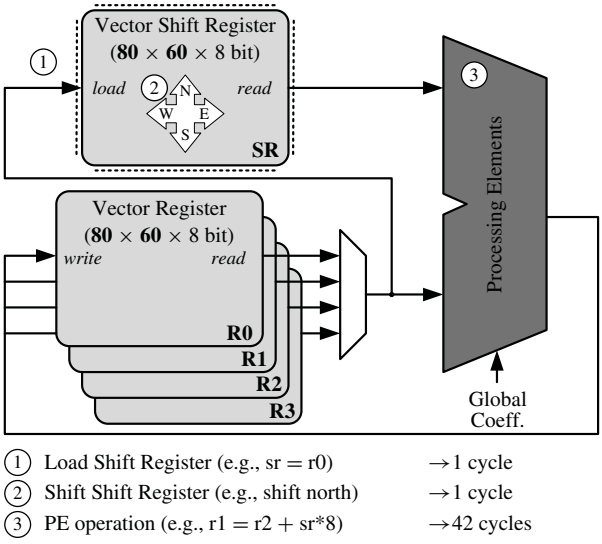


Fig. 12. Simplified programmer's model of the VAE.

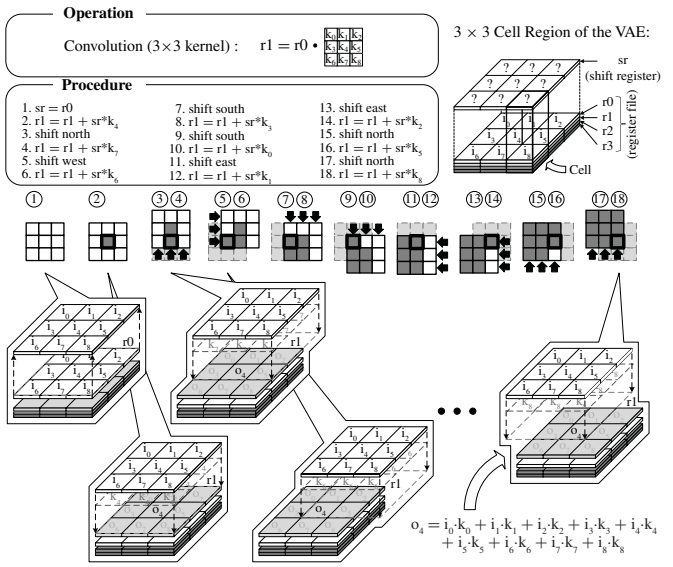


Fig. 13. Procedure for convolution with a 3 × 3 kernel.

cycle. The third operation type, i.e., the PE operation, takes up to three operands (one of r0 ∼ 3, sr, and a global coefficient), performs a PE operation, and saves the result back to r0 ∼ 3. This takes 42 cycles to complete as a result of the pipelined PE operation explained earlier.

A. CNN Operation

Using vector notation, a 3 × 3 CNN template T, which defines the behavior of a CNN, can be expressed as

$$\mathrm{T} = \{\mathrm{A}, \mathrm{B}, i\} \tag{5}$$

where $\mathrm{A} = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \end{bmatrix}$ and $\mathrm{B} = \begin{bmatrix} b_0 & b_1 & b_2 \\ b_3 & b_4 & b_5 \\ b_6 & b_7 & b_8 \end{bmatrix}$.

Letting U equal the 80 × 60-dimensional input image matrix and X(k) equal the 80 × 60-dimensional CNN state matrix at time k, one time step in the CNN Euler approximation (3) applied to the entire 80 × 60 array can be expressed as

$$\mathrm{X}(k+1) = \mathrm{X}(k) * \mathrm{A} + \mathrm{U} * \mathrm{B} + i. \tag{6}$$

Although (4) originally required an additional saturation step, this step is eliminated by assuming the FSR model, in which the saturation function is always applied to the PE output, and thus $y^c = x^c$.

Equation (6) shows that CNN operation is mainly composed of a 2-D convolution with a kernel. Fig. 13 visualizes an efficient procedure for calculating the convolution, which involves a spiral shifting motion to minimize data access overhead and maximize PE utilization. First, the register array holding the input is copied into the shift register array. Then the shift register array is shifted in a spiraling motion. From the viewpoint of the center cell, the state value of each of its neighbor cells becomes available after each shift operation. The convolution is obtained by multiplying the shift register value with the appropriate coefficient and accumulating the result back to the result after each shift. Thanks to the efficient shift pattern and single cycle shift operations, data communication overhead is only 2.4% and utilization of the PE pipelines is 93%. Convolution using larger kernels (5 × 5, 7 × 7, and so on) can also be efficiently achieved by simply extending the spiraling shift sequence.

According to (6), a CNN iteration is completed by executing two convolutions and one addition. First, a convolution is calculated for the state output X(k) and is repeated for the input U. Then the threshold value i, which is normally a negative value, is added to the state to obtain the final state X(k+ 1). For a 3 × 3 CNN template with all nonzero coefficients, this process takes 858 cycles or 4.3 μs at 200 MHz.

B. VA

The saliency-based VA requires Gaussian filtering, Gabor-like filtering, and center-surround operations. Gabor-like filtering, unlike the other operations, requires a complex-valued CNN and is examined in more detail here. The CNN template for a Gabor-like filter is given by

$$\mathrm{A} = \begin{bmatrix} 0 & e^{-j\omega_{yo}} & 0 \\ e^{j\omega_{xo}} & -(4+\lambda^2) & e^{-j\omega_{xo}} \\ 0 & e^{j\omega_{yo}} & 0 \end{bmatrix}$$
$$\mathrm{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and } i = 0. \tag{7}$$

Here, ω_{xo} and ω_{yo} are the angular frequencies of the Gabor-like filter orthogonal to the x and y axes, respectively, and λ is a constant that is inversely proportional to the cut-off distance of the low pass envelope of the Gabor-like filter.

The matrix A contains both real and imaginary parts. Letting $\mathrm{A}_{\mathrm{real}}$ and $\mathrm{A}_{\mathrm{imag}}$ equal the value of the real and imaginary parts of A, then (6) can be transformed into

$$\mathrm{X}_{\mathrm{real}}(k+1) = \mathrm{X}_{\mathrm{real}}(k) * \mathrm{A}_{\mathrm{real}} - \mathrm{X}_{\mathrm{imag}}(k) * \mathrm{A}_{\mathrm{imag}} + \mathrm{U} * \mathrm{B} \tag{8}$$

and

$$\mathrm{X}_{\mathrm{imag}}(k+1) = \mathrm{X}_{\mathrm{real}}(k) * \mathrm{A}_{\mathrm{imag}} + \mathrm{X}_{\mathrm{imag}}(k) * \mathrm{A}_{\mathrm{real}}. \tag{9}$$

TABLE I
EXECUTION TIME SUMMARY

Operation	Execution time (μs)	Percentage of total
Gaussian Filter	521	21%
Gabor-type Filter	1210	50%
Center-Surround Filter	326	13%
Other	383	16%
Total	2440	

TABLE II
CHIP IMPLEMENTATION SUMMARY

Process technology	0.13-μm eight metal CMOS
Area	4.5 mm^2
Number of cells	4800 (80×60)
Number of PEs	120
Operating voltage	1.2-V
Operating frequency	200 MHz
Peak performance	24 GOPS
Sustained performance	22 GOPS
Active power consumption	84 mW

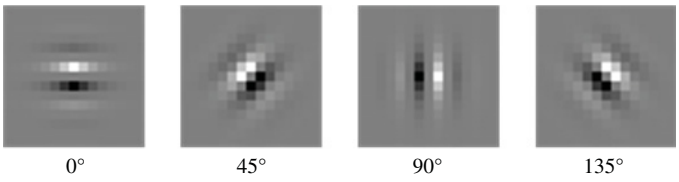


Fig. 14. Gabor-type filter impulse response (imaginary component).

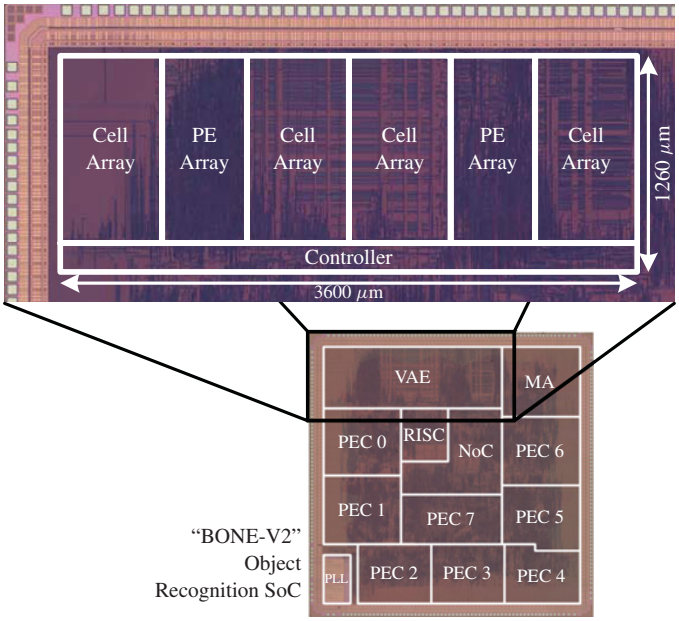
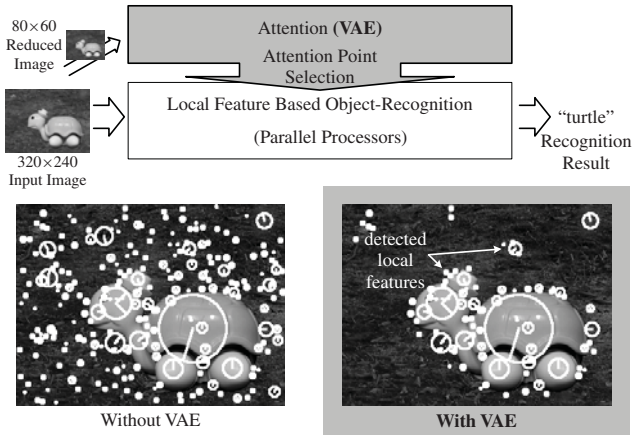


Fig. 15. Chip photograph.



	Without VAE	With VAE	Improvement
Number of Extracted Local Features	279	**100**	**64% ↓**
Recognition Frame Rate	12	**22**	**83% ↑**
Energy per Frame	42 mJ	**23 mJ**	**45% ↓**

Fig. 16. Object-recognition results.

Although (8) and (9) seem considerably more complex than (6), thanks to the large number of zero entries in the template, the number of required MAC operations for one step of the Gabor-type filter template is only 19, which is actually equal to that of (6). Only the storage requirement is increased, since all four registers r0 ∼ r3 are required to compute the complex CNN. As a result, one iteration of the Gabor-type filter takes 831 cycles or 4.15 μs, which is less than (6) due to omission of the threshold addition. The impulse responses of Gabor-type filters with varying orientation are shown in Fig. 14. In all cases, the angular frequency ω is equal to 1.5 radians, and λ is equal to 0.66, which corresponds to a 6-dB cutoff distance of 1.5 pixels. Fifteen iterations were used in each case, resulting in 65 μs execution time.

Table I summarizes the operations performed by the VAE and the time spent on each operation. Gabor-type filtering, which is used for extracting specific edge orientations, takes more than half of the total execution time, due to the large number of iterations required for each operation.

VI. IMPLEMENTATION RESULTS

The 4.5-mm^2 VAE was fabricated as part of the 36-mm^2 object-recognition SoC [7] shown in Fig. 2 using a 0.1-μm eight metal logic CMOS technology. The cell arrays are custom-designed to minimize the area while the PE and controller blocks are synthesized. The photograph of the resulting chip is shown in Fig. 15, and the chip features are summarized in Table II. Power consumption is 84 mW when running at 200 MHz and 1.2-V. The 120 PEs have a peak performance of 24 GOPS and show a utilization rate of 93% during CNN operation, thereby sustaining an effective performance of 22 GOPS.

Fig. 16 shows the result of applying VAE to an object-recognition SoC designed for real-time (>15 frames per second) operation on 320 × 240 video input. The VAE rapidly performs a saliency-based attention algorithm on the 80 × 60 scaled image and outputs a pixel map marking the ROIs. This map is used later by the parallel processors of the object-recognition SoC to perform detailed local-feature-based object-recognition only on regions of high saliency. For object images with background clutter, the average number of local

features is drastically reduced when the VAE is activated, increasing frame rate by 83% and reducing energy per frame by 45% without degrading the recognition rate. The energy overhead of the VAE itself is only 0.2 mJ/frame thanks to the short processing time of 2.4 ms/frame.

VII. Conclusion

In this paper, we have presented a CNN-based hardware acceleration block specialized for executing the VA algorithm. Named the VAE, it adopts the TMPE topology to achieve sustained performance of 22 GOPS while consuming just 84 mW. This enables it to complete a CNN iteration on an 80×60 pixel image in 4.3 μs and thus complete the VA algorithm in just 2.4 ms. By applying the VAE, object-recognition speed is increased by 83% with no negative effects on recognition accuracy.

References

[1] D. Kim, K. Kim, J.-Y. Kim, S. Lee, and H.-J. Yoo, "An 81.6 GOPS object recognition processor based on NoC and visual image processing memory," in *Proc. IEEE Custom Integr. Circuits Conf.*, San Jose, CA, Sep. 2007, pp. 443–446.

[2] S. Kyo, S. Okazaki, T. Koga, and F. Hidano, "A 100 GOPS in-vehicle vision processor for pre-crash safety systems based on a ring connected 128 4-way VLIW processing elements," in *Proc. IEEE Symp. VLSI Circuits*, Honolulu, HI, Jun. 2008, pp. 28–29.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[4] S. Lee, K. Kim, M. Kim, J.-Y. Kim, and H.-J. Yoo, "The brain mimicking visual attention engine: An 80×60 digital cellular neural network for rapid global feature extraction," in *Proc. IEEE Symp. VLSI Circuits*, Honolulu, HI, Jun. 2008, pp. 26–27.

[5] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[6] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annu. Rev. Neurosci.*, vol. 18, pp. 193–222, Mar. 1995.

[7] K. Kim, S. Lee, J.-Y. Kim, M. Kim, D. Kim, J.-H. Woo, and H.-J. Yoo, "A 125GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual-attention engine," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2008, pp. 308–615.

[8] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.-I*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988.

[9] B. E. Shi, "Gabor-type filtering in space and time with cellular neural networks," *IEEE Trans. Circuits Syst. I*, vol. 45, no. 2, pp. 121–132, Feb. 1998.

[10] C. Koch and S. Ullman, "Shifts in selective visual attention: Toward the underlying neural circuitry," *Hum. Neurobiol.*, vol. 4, no. 4, pp. 219–227, 1985.

[11] L. Itti and C. Koch, "A comparison of feature combination strategies for saliency-based visual attention systems," in *Proc. SPIE Conf. Hum. Vis. Electron. Imaging IV*, San Jose, CA, Jan. 1999, pp. 473–482.

[12] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Washington D.C., Jun.–Jul. 2004, pp. 37–44.

[13] A. Rodríguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, S. E. Meana, "ACE16k: The third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 5, pp. 851–863, May 2004.

[14] P. Kinget and M. S. J. Steyaert, "A programmable analog cellular neural network CMOS chip for high speed image processing," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 235–243, Mar. 1995.

[15] P. Keresztes, Á. Zarándy, T. Roska, P. Szolgay, T. Bezák, T. Hidvégi, P. Jónás, and A. Katona, "An emulated digital CNN implementation," *J. VLSI Signal Process. Syst.*, vol. 23, nos. 2–3, pp. 291–303, Nov.–Dec. 1999.

[16] M. Laiho, A. Paasio, A. Kananen, and K. A. I. Halonen, "A mixed-mode polynomial cellular array processor hardware realization," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 2, pp. 286–297, Feb. 2004.

[17] P. Földesy, A. Zarándy, and C. Rekeczky, "Configurable 3-D-integrated focal-plane cellular sensor-processor array architecture," *Int. J. Circuit Theory Appl.*, vol. 36, nos. 5–6, pp. 573–588, Jul.–Sep. 2008.

[18] S. Espejo, R. Carmona, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "A VLSI-oriented continuous-time CNN model," *Int. J. Circuit Theory Appl.*, vol. 24, no. 3, pp. 341–356, May–Jun. 1996.

Seungjin Lee (S'06) received the B.S. and M.S. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006 and 2008, respectively. He is currently working toward the Ph.D. degree in electrical engineering and computer science at KAIST.

He joined the Semiconductor System Laboratory at KAIST as a Research Assistant in 2006. His current research interests include parallel architectures for computer vision processing, low-power digital signal processors for digital hearing aids, and body area communication.

Minsu Kim (S'07) received the B.S. and M.S. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2007 and 2009, respectively.

He is currently with KAIST. His current research interests include network-on-chip based system-on-chip design and bio-inspired very-large-scale integration architecture for intelligent vision processing.

Kwanho Kim (S'04–M'09) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2004, 2006, and 2009, respectively.

He is currently with the Digital Media & Communication Research and Development Center, Samsung Electronics, Gyeonggi-do, Korea. His current research interests include very-large-scale integration design for object recognition, architecture, and implementation of network-on-chip-based system-on-chip.

Joo-Young Kim (S'05–M'10) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2005, 2007, and 2010, respectively.

He has been with KAIST and involved with the development of parallel processors for computer vision since 2006. His current research interests include parallel architecture, subsystems, and very-large-scale integration implementation for bio-inspired vision processors.

Hoi-Jun Yoo (M'95–SM'04–F'08) received the Graduate degree from the Electronic Department, Seoul National University, Seoul, Korea, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1985 and 1988, respectively. His Ph.D. work concerned the fabrication process for GaAs vertical optoelectronic integrated circuits.

He was with Bell Communications Research, Red Bank, NJ, from 1988 to 1990, where he invented the 2-D phase-locked vertical-cavity surface-emitting laser array, the front-surface-emitting laser, and the high-speed lateral heterojunction boiler transistors. In 1991, he became a Manager of the Dynamic Random Access Memory (DRAM) Design Group, Hyundai Electronics, Kyoungki-do, Korea, and designed a family of fast 1M DRAMs and 256M synchronous DRAMs. In 1998, he joined the faculty of the Department of Electrical Engineering, KAIST, and is now a Full Professor. From 2001 to 2005, he was the Director of the System Integration and IP Authoring Research Center, Daejoen, funded by the Korean government to promote worldwide IP authoring and its system-on-chip (SoC) to the Minister in the Ministry of Information and Communication, Korea, and National Project Manager for SoCs and Computers. In 2007, he founded the System Design Innovation & Application Research Center at KAIST to research on and develop SoCs for intelligent robots, wearable computers, and bio systems. He has authored two books: *DRAM Design* (Seoul, Korea: Hongleung, 1996, in Korean) and *High-Performance DRAM* (Seoul, Korea: Sigma, 1999; in Korean) and wrote chapters for *Networks-on-Chips* (New York: Morgan Kaufmann, 2006). His current research interests include high-speed and low-power network on chips, 3-D graphics, body area networks, biomedical devices and circuits, and memory circuits and systems.

Prof. Yoo is the Technical Program Committee Chair of the Asian Solid-State Circuits Conference (A-SSCC 2008). He is the recipient of the Electronic Industrial Association of Korea Award for his contribution to DRAM technology in 1994, the Hynix Development Award in 1995, the Korea Semiconductor Industry Association Award in 2002, the Best Research of KAIST Award in 2007, the Design Award of the Asia and South Pacific Design Automation Conference in 2001, and the Outstanding Design Awards of A-SSCC in 2005, 2006, and 2007. He is a member of the executive committee of the International Solid-State Circuits Conference, the Symposium on Very-Large-Scale Integration Design, and the A-SSCC.