IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 17, 2018, accepted June 22, 2018, date of publication July 4, 2018, date of current version August 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2852809

Lossy Compression for Embedded Computer Vision Systems

LI GUO[1], DAJIANG ZHOU[1], (Member, IEEE), JINJIA ZHOU[2,3], (Member, IEEE), SHINJI KIMURA[1], (Member, IEEE), AND SATOSHI GOTO[1], (Life Fellow, IEEE)

[1]Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan
[2]School of Science and Engineering, Hosei University, Tokyo 184-8485, Japan
[3]JST, PRESTO, Tokyo 102-0076, Japan

Corresponding author: Li Guo (guoli@toki.waseda.jp)

This work was supported in part by Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists and in part by JST, PRESTO, Japan, under Grant JPMJPR1757.

ABSTRACT Computer vision applications are rapidly gaining popularity in embedded systems, which typically involve a difficult tradeoff between vision performance and energy consumption under a constraint of real-time processing throughput. Recently, hardware (FPGA and ASIC-based) implementations have emerged, which significantly improves the energy efficiency of vision computation. These implementations, however, often involve intensive memory traffic that retains a significant portion of energy consumption at the system level. To address this issue, we are the first researchers to present a lossy compression framework to exploit the tradeoff between vision performance and memory traffic for input images. To meet various requirements for memory access patterns in the vision system, a line-to-block format conversion is designed for the framework. Differential pulse-code modulation-based gradient-oriented quantization is developed as the lossy compression algorithm. We also present its hardware design that supports up to 12-scale 1080p@60fps real-time processing. For histogram of oriented gradient-based deformable part models on VOC2007, the proposed framework achieves a 49.6%–60.5% memory traffic reduction at a detection rate degradation of 0.05%–0.34%. For AlexNet on ImageNet, memory traffic reduction achieves up to 60.8% with less than 0.61% classification rate degradation. Compared with the power consumption reduction from memory traffic, the overhead involved for the proposed input image compression is less than 5%.

INDEX TERMS Computer vision, feature extraction, lossy compression, memory traffic reduction.

I. INTRODUCTION

Computer vision algorithms have been evolving rapidly and gaining popularity in embedded devices, including smartphones and driver assistance systems [1]. More applications are emerging, such as vision on wireless sensor networks. Whereas many of these applications are battery powered or even battery-less [2], [3], low energy consumption is crucial for embedded systems with limited energy resources. To ensure satisfactory vision performance, however, high-complexity vision algorithms, such as deep neural networks, together with high processing throughput in terms of resolution and frame rate are often desirable, which critically challenges low-power and real-time implementations.

Recently, computer vision hardware implementations have emerged to accelerate processing and reduce power consumption on platforms including FPGA [4], ASIC [5], [6], and a combination of the two [7]. The widely applied object detection algorithm, using the histogram of oriented gradients (HOG) [8] descriptor in combination with a support vector machine (SVM) for classification, has been implemented in FPGA [4] and ASIC [6]. Hahnle *et al.* [4] presented real-time 18-scale pedestrian detection for 1080HD video at 64fps. An energy-efficient ASIC implementation was presented by Suleiman and Sze [6] that supported multi-scale detection at 1080HD 60fps. For more complex deep learning algorithms, hardware implementations of convolutional neural networks (CNNs) have also been presented. Chen *et al.* [10] introduced a CNN implementation in ASIC that achieved a speedup of 450.65× over a GPU. In addition to their hardware computation cores, all these implementations involve intensive memory traffic that composes a significant portion of power consumption at the system level.

The energy efficiency of the computation can be improved by replacing software processing with hardware.

To further reduce power consumption, some implementations also reduce computational complexity through a tradeoff between hardware cost (power and area) and vision performance. However, the power consumption for memory traffic remains a bottleneck, in particular for external DRAM traffic. Considering the HOG/SVM-based ASIC processor [6] as an example, even if scale generation architecture was developed to ensure full reuse in each input frame, reading these frames on a DDR3-1333 interface corresponds to a power dissipation of 96.6 mW (see Section V). The power consumption of memory traffic is clearly dominant compared with the computation core of the detector that consumes 45.3 mW [6]. Hence, for energy-constrained embedded systems, it is essential to reduce memory traffic and its power consumption.

Memory traffic for computer vision applications mainly contains three components: input images, feature maps, and weights. For a system on chip (SoC) that targets a wider range of application scenarios, input images are more likely to be shared by multiple components, such as the vision processor, video encoder, and image processors for denoising/enhancement. The processing speed of these components can also be different, which leads to various amounts of delay between the sensor and components. As a result, a DRAM-based buffer, which is sufficiently large to store several frames of high-resolution video and is sharable through an on-chip bus, can be a more practical choice for such a multi-purpose SoC. In this case, the input images of the vision processor can be obtained directly from sensors or read from DRAM. For the former, a large on-chip SRAM is required to buffer multiple lines of images, in particular for multi-scale processing and high-resolution images. Thus, reading from DRAM is a reasonable choice. Feature maps and weights depend on different algorithms and implementations. For hand-crafted feature-based vision and small-scale CNNs, both models and intermediate results (e.g., feature maps and weights) can be directly stored in the on-chip memory, which does not involve memory traffic from external DRAM. As an example, the common structure of ASIC implementations [6] for HOG/SVM and [11] for scale-invariant feature transform (SIFT)-based object detectors are shown in Fig. 1. Both extracted HOG/SIFT feature maps and weights for detection are stored in on-chip SRAM. Thus, the input images become dominant for external memory traffic.

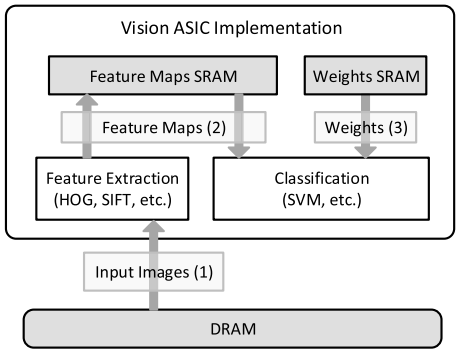

FIGURE 1. Memory traffic for the ASIC implementation of vision applications.

We study the feasibility of reducing the memory traffic of input images using compression (IIC). IIC performs an on-line compression of images before storing them in the DRAM, and the corresponding decompression after they are fetched. We choose IIC from the memory traffic optimization techniques based on the following reasons: 1) the design of IIC is independent of computer vision algorithms; hence, one IIC can be generally used in almost all algorithms; and 2) IIC can be flexibly combined with other optimizations for memory traffic reduction, such as data caching, reusing [12], and scale generation architecture [6].

IIC can be either lossless or lossy; however, we focus on the latter to explore the maximum memory traffic reduction at an acceptable detection/classification performance, although a lossless IIC is also given as a baseline for comparison. Vision-oriented lossy IIC has clear potential to benefit from quantization optimized for a vision-related criterion (e.g., minimizing the error in gradients) rather than simply following a conventional visual experience-oriented cost function, such as the peak signal-to-noise ratio (PSNR).

For a complete IIC framework, its transparency to data users should also be addressed. Unlike in video codec systems [13] where re-compression for reference frames (i.e. IIC) is exclusively used by the encoder or decoder core, IIC in a vision application is most likely to be shared by multiple components that access the image data in various manners, including the vision core, image sensor, and/or preprocessor. Therefore, it is preferable that the IIC algorithm supports the extraction of compressed data in different scanning orders.

The contributions of this work are summarized as follows:

1) a new tradeoff between vision accuracy drop and power consumption reduction: whereas many previous works studied the tradeoff between vision accuracy and energy consumption of the vision core, this work provides a new angle of view by replacing the latter with memory traffic, which achieves better efficiency;
2) an efficient processing format conversion to support flexible input/output scanning orders: compared with the conventional line buffer solution, the proposed approach reduces the buffer size by 72.9–91.3%;
3) a gradient-oriented lossy IIC algorithm: by optimizing quantization toward the minimization of error of gradients rather than the conventional PSNR, the proposed lossy IIC achieves 10%–15% better memory traffic reduction at a comparative detection accuracy;
4) hardware implementation for the proposed lossy IIC: it achieves a throughput of 1.6 Gpixels/s that can support real-time detection at 1080HD 60fps with 12 image scales for detecting various sizes of objects (e.g. [6]). Relative to the saved memory power, the power overhead introduced by IIC is less than 5%.

The remainder of this paper is organized as follows: An overview of the proposed IIC framework is introduced

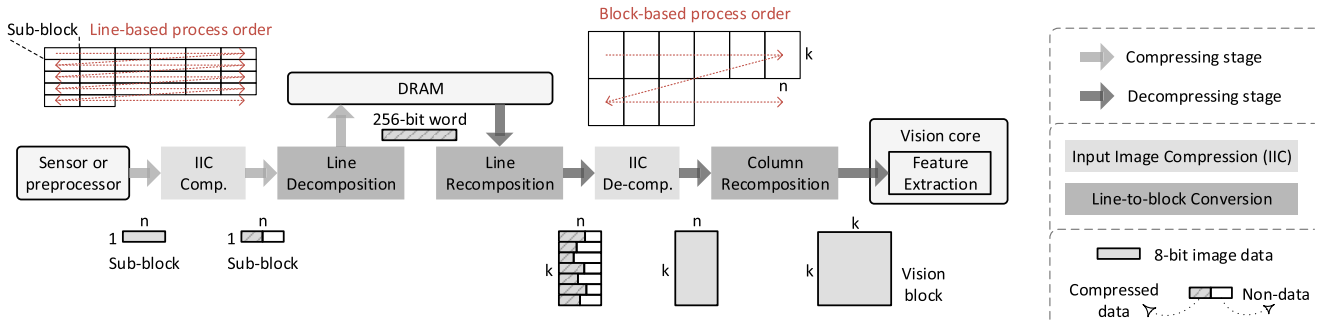

FIGURE 2. Block diagram and data pattern of the proposed input image compression framework.

in Section II. In Section III, a new lossy compression for computer vision and a low-cost processing format conversion are presented. The compression algorithm is based on the widely applied differential pulse-code modulation (DPCM) prediction [14] and proposed gradient-oriented quantization (GOQ). In Section IV, the hardware architecture for the proposed IIC core is presented. Compression performance and vision accuracy are evaluated and given in Section V together with the hardware implementation results. Finally, conclusions are drawn in Section VI.

II. OVERVIEW OF THE FRAMEWORK

A. DESIGN CHALLENGES

The design of a lossy IIC framework for computer vision applications should address the following aspects:

1) For lossy IIC, it is important to determine a balance among memory traffic, vision performance (e.g., detection or classification accuracy), and hardware cost.
2) IIC is expected to support multiple input/output scanning orders as shown in Fig. 3. The input images provided by image sensors or preprocessors are typically line-based, whereas block-based outputs are required for feature extraction, such as HOG, in the vision core. A line buffer is a straightforward solution, but the buffer size is proportional to the image resolution, which leads to a huge hardware cost.
3) Existing lossy image compression algorithms (e.g., for video codec) are typically optimized for the human visual experience rather than computer vision, which focuses on minimizing the errors of the pixel map. For vision algorithms, however, errors in the gradient map are more important.

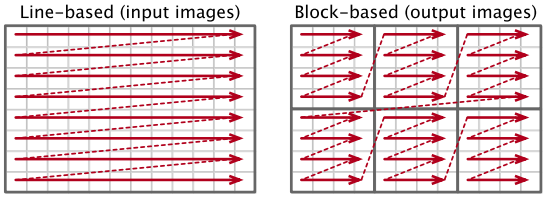

FIGURE 3. Typical scanning orders of input/output images.

In this work, a vision-oriented lossy IIC framework is presented. In Section III.A, a lossy compression algorithm to minimize the error in gradients rather than the PSNR is introduced. To support flexible input/output scanning orders, a low-cost processing format conversion is proposed in Section III.B. Finally, experimental results, and a trade-off between memory traffic and vision performance are given in Section V.C.

B. LOSSY INPUT IMAGE COMPRESSION FRAMEWORK

As shown in Fig. 2, the proposed lossy IIC framework is composed of a lossy IIC core and processing format conversion. Corresponding to the compressor and decompressor in an IIC core, the line-to-block conversion contains a line decomposition process and block recomposition process (including line and column recomposition). The compression stage follows the line-based scanning order, whereas the decompression stage follows the block-based raster order.

Two types of blocks are related to this IIC work. One is used in the design of the IIC algorithm and is defined as a compression block. With a size of $n \times m$, it breaks the data dependency between blocks so that part of an image can be obtained without reading all the previous lines. The compression block is further divided into $n \times 1$ sub-blocks for the line-based compression process. The other is the block of $k \times k$ in the vision processor, which is viewed as a vision block. The two types of blocks serve different purposes and do not have to be equal in size. For each process of compression and conversion, the data patterns are also described in Fig. 2.

In the compression stage, line-based images from image sensors or preprocessors are first divided into $n \times 1$ sub-blocks. These sub-blocks are compressed and stored line-by-line in the external memory. After the line-based compressed data are retrieved from DRAM, they are converted to $n \times k$ blocks by the line recomposition portion. Then the block-based data are decompressed in the IIC core. Finally, the restored 8-bit $n \times k$ image blocks are further converted to $k \times k$ vision blocks and exported to feature extraction in the vision core.

The block diagram of the IIC decompression process between feature extraction and the DRAM interface is shown in Fig. 4. The block request from feature extraction is first

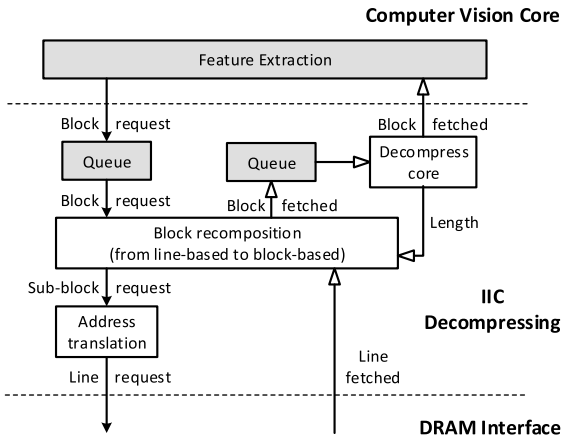

FIGURE 4. Basic structure of the IIC decompression process.

checked by buffers in the block recomposition. Unless all coded sub-blocks in the requested block are stored in buffers, the request for missing sub-blocks is sent to the DRAM interface. During the compression process, sub-blocks are compressed to a variable length, and then they are merged and decomposed into words for storage. Thus, to read a sub-block from external memory in the decompression process, the address translation unit needs to translate the sub-block requests into word requests for the DRAM interface. After all sub-blocks are retrieved, they are merged into a block and decompressed. Finally, the restored image blocks are returned to feature extraction. To reduce DRAM access latency, FIFO queues are inserted between the functional blocks to pipeline the entire process.

Additionally, the proposed IIC framework is suitable for both lossless and lossy compression. Lossless IICs are important for systems with error propagation, such as video coding. Better compression performance can be achieved by lossy IICs with a slight image quality loss. Without error propagation and frame feedback, this small error does not accumulate and become a severe problem. Because there is no image feedback in computer vision applications, lossy compression is preferred for a larger reduction in memory traffic. In our work, DPCM prediction and GOQ lossy coding are combined, as presented in the following section.

III. PROPOSED INPUT IMAGE COMPRESSION

A. LOSSY INPUT IMAGE COMPRESSION ALGORITHM

1) OVERALL PROCESSING FLOW OF THE IIC CORE

Some previous IIC studies have been conducted for video codecs [14]–[19], [21] and display systems [20]. The compression is typically composed of prediction and entropy coding. Techniques for the prediction phase can be approximately divided into two types: spatial [14]–[19] and frequency [20], [21] domain prediction. Because of low computational complexity, prediction in the spatial domain is applied more frequently, including DPCM scanning [14], and hierarchical average and copy [16]. For the entropy coding phase, variable length coding is widely used, such as Exp-Golomb Rice coding [18], significant bit truncation (SBT) [16], and semi-fixed length coding [14].

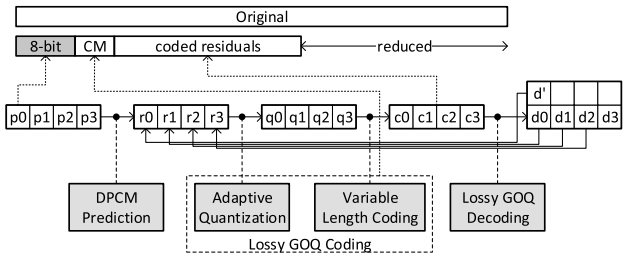

FIGURE 5. Overall processing flow of the proposed DPCM-based lossy GOQ coding algorithm.

Following the basic structure of previous IICs [14]–[19], we proposed a lossy IIC [22] which contains three portions, as shown in Fig. 5: DPCM prediction, lossy GOQ coding, and decoding. To improve compression performance, multi-mode prediction [15] and prediction between several rows [16] are applied. However, if prediction in the vertical direction increases, then the hardware cost of processing format conversion is higher. Therefore, it is a tradeoff between hardware cost and compression performance.

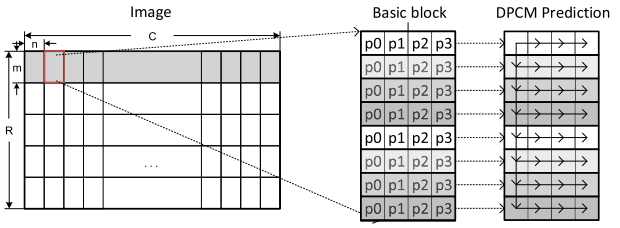

FIGURE 6. DPCM prediction in one block.

To reduce the prediction between rows, DPCM scanning is used, as shown in Fig. 6. Because all the sub-blocks are processed in order and there is no random access, vertical DPCM prediction within one block is performed to further reduce the presenting bits. The top-left pixel in a block retains its original 8-bit value (F). In Fig. 5, p_0, p_1, p_2, and p_3 are the original 8-bit values of the input image inside a sub-block. After prediction, the obtained residuals are r_0, r_1, r_2, and r_3. d_0, d_1, d_2, and d_3 are the decoded values of corresponding pixels. d' is the decoded value of p_0's upper pixel.

In the lossy coding stage, a GOQ coding method is introduced for vision applications. Residuals from the DPCM prediction are first quantized as q_0, q_1, q_2, and q_3. Based on the variable length coding of SBT [16], these quantized residuals are further coded to be the same length, and an overhead of coding mode (CM) is added to indicate the coded bit length (see Section III.A.2).

In the lossy decoding portion, the decoded pixels' values (d_0, d_1, d_2, and d_3) are calculated for DPCM prediction. This can reduce the propagation error caused by prediction with the original pixels and lossy coding. Fig. 7 shows an example

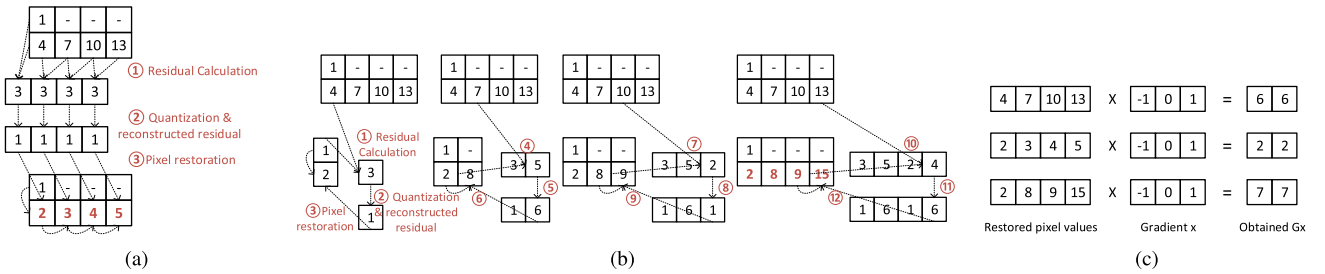

FIGURE 7. Comparison between the prediction, and original pixel and decoded pixel values: (a)(b) restored pixel values with the original and decoded pixels, respectively; (c) gradient magnitude comparison in the horizontal direction.

of prediction with the original and restored pixels. Gradient information plays a more important role than the pixel values in vision algorithms; hence, the gradient magnitude in the horizontal direction is displayed in Fig. 7c. Compared with prediction with the original pixels, the decoded pixels achieve much smaller errors in both pixel and gradient maps.

Finally, the 8-bit top-left pixel, CM, and coded residuals are merged as a compressed bit stream for the output (viewed as S in Section III.B). This is stored in the external DRAM instead of the original 8-bit image.

2) GRADIENT-ORIENTED QUANTIZATION CODING

GOQ lossy coding involves a GOQ and variable length coding. The residuals obtained from DPCM prediction are first quantized based on the variable gradient-oriented quantization coefficients (QC). Then a simple variable length coding of SBT [16] is performed.

Linear quantization is widely used in conventional lossy coding. It is efficient for applications evaluated by the visual experience, such as video coding. Video coding is used for the storage or transmission of video sequences; hence, minute differences between neighboring pixels cannot be noticed by human eyes and can be ignored for compression. However, vision algorithms focus more on the gradient-based information, that is the difference between values of neighboring pixels. Hence, even small differences should be retained to guarantee the accuracy of the vision algorithm, and GOQ should be used instead of linear quantization.

Fig. 8 shows an example of different influences of linear residual quantization on video coding and vision algorithm. For video coding, the restored block is obtained by adding residuals to the pixel value used for prediction. This small difference cannot be detected by human eyes, so linear quantization is efficient for applications evaluated by a visual experience. However, for vision algorithms, both gradient magnitudes and directions change. Some of them are even zero after linear quantization, which decreases the detection performance of vision algorithms. Hence, conventional linear quantization is not suitable for vision applications.

To reduce the effect of quantization on the gradient-based feature, smaller differences between the neighboring pixels should be retained. Therefore, residuals with smaller values should be quantized by smaller QC. Based on this principle,

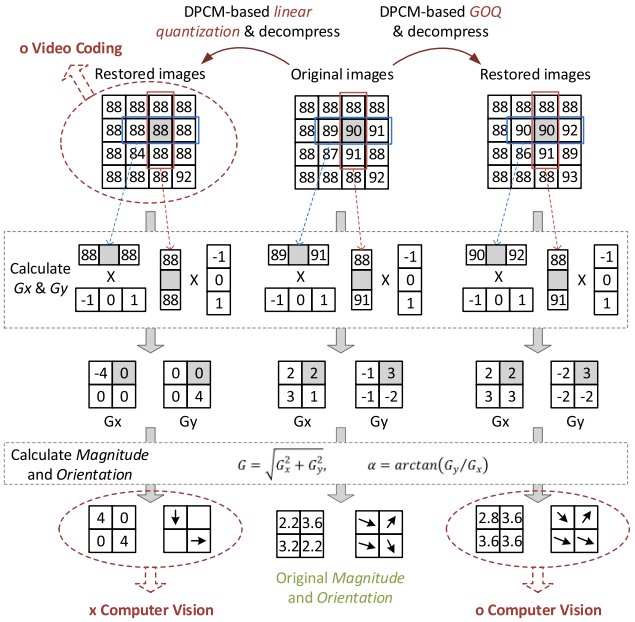

FIGURE 8. Differences between video coding and gradient-based feature extraction. (for linear quantization, all residuals are quantized by four.)

an example of the GOQ for vision algorithms is shown in Table 1.

TABLE 1. Example of gradient-oriented quantization.

QC	r	q	QC	r	q
1	0	0	5	±(32 - 36)	±8
3	±(1 - 3)	±1	7	±(37 - 43)	±9
3	±(4 - 6)	±2	7	±(44 - 50)	±10
5	±(7 - 11)	±3	7	...	...
5	...	...	7	±(86 - 92)	±16

Note: "QC" denotes the quantization coefficient, r is the residual value, and q is the quantized residual value.

For variable length coding, the basic concept of SBT [16] coding is to present the residual with as many truncated significant bits as possible. All the residuals within a sub-block are coded with the same bit length. Based on the range of quantized residuals (q) in a sub-block, a CM is determined.

TABLE 2. Variable length coding for lossy GOQ coding.

CM	Bit Num.	Range of q	Range of r	Prob. (%)	CM code
A	0	0	0	2.84	00000
B	1	[-1, 0]	[-3, 0]	2.16	00001
C	2	[-2, 1]	[-6, 3]	22.83	01
D	3	[-4, 3]	[-16, 11]	31.00	10
E	4	[-8, 7]	[-36, 31]	23.28	11
F	5	[-16, 15]	[-92, 85]	14.18	001
G	5	Others, pixel quantized by 8		3.71	0001

Note: q is the quantized residual value and r is the residual value.

Table 2 shows the relationship between the CM, range of the quantized residual, and range of the residual based on QC, as shown in Table 1. If the range of quantized residuals is larger than ± 16, that is, in the CM G, all the original pixels in this sub-block are quantized by eight instead of residuals.

The proposed GOQ lossy coding is tested on the VOC2007 trainval dataset (detailed experimental conditions are presented in Section V). The probabilities of each CM are shown in Table 2. According to Huffman coding, fewer bits are used to present a CM with a large probability.

Additionally, both the GOQ in Table 1 and linear quantization by four are evaluated on VOC2007. The average PSNR of both the restored pixel map and gradient map are shown in Table 3. For similar pixel PSNRs (0.2 dB difference), the proposed GOQ coding achieves a much higher gradient PSNR. Its gradient PSNR improves by 1.05dB, which implies that the average gradient error of GOQ reduces by 11.2x compared with linear quantization. The detailed experimental results and analysis of the detection/classification accuracy are presented in Section V.C.

TABLE 3. Average PSNR comparison between linear and gradient-oriented quantization (GOQ).

	Linear Quantization	GOQ
DRR[1](%)	49.83	49.63
PSNR (pixel map, dB)	49.6982	49.8987
PSNR (gradient map, dB)	45.0159	46.0647

[1] *data reduction ratio*

B. PROCESSING FORMAT CONVERSION

To support multiple input/output scanning orders, a low-cost format conversion is located between the external memory and IIC core. Together with the required data formats in each processing step, the format conversion is shown in Fig. 9, which consistes of line decomposition and block recomposition.

Flexible format conversion is added to address the following three issues: 1) After compression, sub-block ($n\times1$) is encoded to a variable length, so the line decomposition portion is added to map them into fixed-length words for storage. 2) Because of the variable compressed size, it is difficult to locate the required compressed sub-block in DRAM during decompression. Thus, without causing too much overhead, efficient address generation is implemented. 3) To convert the line-based input to block-based output ($k\times k$), block recomposition is applied, which divides this process into two steps: from $n\times1$ to $n\times k$ for decompression and then to $k\times k$ for feature extraction.

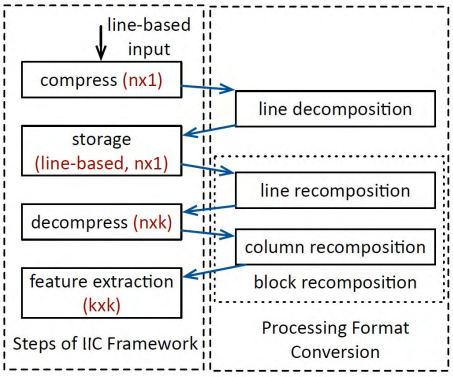

FIGURE 9. Overview of the required data format in each processing step of the IIC framework and processing format conversion.

1) LINE DECOMPOSITION

For one line of the input image, line decomposition and the proposed address generation are described in Fig. 10. The original image is first divided into $n\times1$ sub-blocks (S) and compressed into a variable length. These data are then merged into a bit stream. Finally, the stream is split into 256-bit works and stored in DRAM.

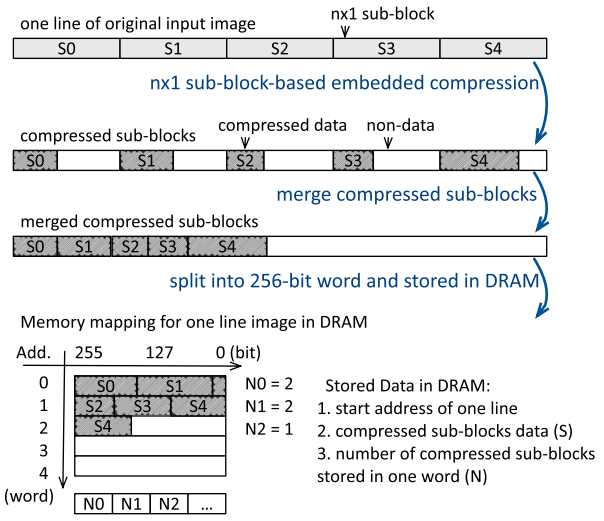

FIGURE 10. Line decomposition and address generation for one line of the input image. If one sub-block is stored in two words, then it is counted in the second.

Because of the variable compressed size and merged storage of the sub-blocks (S), addresses for each sub-block

need to be stored. To reduce redundant data access, these addresses can be replaced by the start address of each line and length of each compressed sub-block, as in [14]. However, in this work, the sub-block size is small (e.g. 4×1), so it leads to a large number of lengths to be stored. Thus, how to reduce this huge overhead is a problem. For vision applications, all sub-blocks/blocks are written/read in a fixed line/block-based raster order. Hence, instead of sub-blocks' addresses or lengths, the number of sub-blocks (N) in one 256-bit word is recorded. The exact length can be obtained during decoding, which does not affect the decompression of the following block.

According to the lossy IIC in Section III.A, the length of the compressed sub-blocks (S) ranges from 5 bits to 27 bits. For 256-bit word access, N is less than 52, so 6 bits are sufficient for presenting each N. Moreover, because of the prediction in Fig. 6, a line buffer that is a quarter of the width of the image resolution (when n=4) is required to store the left-upper pixel for prediction in the next line.

2) BLOCK RECOMPOSITION

As shown in Fig. 9, block recomposition consists of a line (from line-based $n\times1$ to block-based $n\times k$) and column (from $n\times k$ to $k\times k$) recomposition.

Two buffers with the total size of $3k$ words are required in line recomposition, including a k-word buffer for storing N from k lines and a $2k$-word buffer for storing two neighboring S words from each line. If any of N/S words in one line is empty (i.e., all decompressed), the request to read the next N/S word from the same line is sent to DRAM. Because of a fixed starting address of each line, the address of the next S word is only related to the position of the N value in one N word. The request to read an N word is always prior to S words. Thus, before reading an S word, its related N word is always retrieved and ready, and requests for N/S words could be sent contiguously.

Because all $n\times k$ blocks are decoded in fixed raster order, column recomposition is achieved by a buffer with the size of $2k^2$ bytes to store two adjacent $k\times k$ reconstructed blocks.

IV. HARDWARE IMPLEMENTATION

The lossy IIC hardware implementation consists of a compress core and decompress core. Compared with previous lossless works [14]–[16], the main difference is the addition of GOQ. Because the quantization in GOQ coding is not two-based, it can be achieved by just shifting. Hence, several look-up tables (LUTs) are designed to match residuals, quantized residuals, and reconstructed residuals.

Fig. 11 shows a two-stage pipelined architecture for the compress core. One sub-block is processed every cycle. In stage 1, pixels in a block are processed in order from p_0 to p_1. Pixel p_i is first predicted by its restored neighboring pixels value d_{i-1}. Then the obtained residual is quantized to be q_i and decoded to calculate the restored pixel value according to LUT_{rd}. In stage 2, the CM is determined based on the range of quantized residuals q_i, as shown in Table 2.

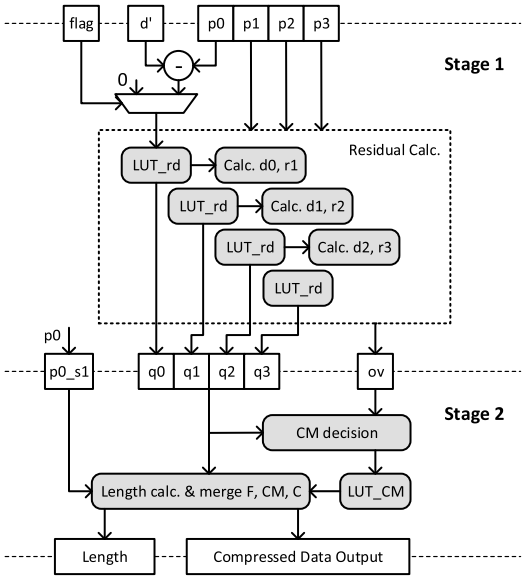

FIGURE 11. Two-stage pipelined compress core architecture.

The CM is coded as per LUT_{CM} presented in Table 2, and merged with p_0 and coded residuals c_i as a bit stream output.

For decompression, a two-stage pipelined architecture is designed, as shown in Fig. 12. Two sub-blocks are decoded every cycle. In stage 1, the compressed image data is first shifted and split into CM and residuals of one sub-block (Sub_r). Then, in stage 2, the merged residuals (Sub_r) are further split into four 5-bit quantized residuals q_i. Using LUT_{ird}, reconstructed residuals are obtained and used to calculate the restored pixel value by inverse DPCM scanning.

The detailed input and output information of LUTs for residual quantization and CM coding are shown in Table 4. These LUTs are designed based on the QC in Table 1 and CM coding in Table 2. Because residual quantization and reconstruction are achieved using LUTs, quantization by different QCs can be easily implemented by replacing LUT_{rd} and LUT_{ird}.

TABLE 4. Look-up table for the compress and decompress cores.

	Input (bit num.)	Output (bit num.)
LUT_{rd}	r_i (9)	q_i (5), rr_i (9)
LUT_{CM}	cm (3)	ccm (4)
LUT_{ird}	q_i (5)	rr_i (9)
LUT_{iCM}	ccm (4)	cm (3)

Note: *r_i is the residual value, q_i is the quantized residual, rr_i is the restored residual during decoding, cm is the coding mode, and ccm is the coded coding mode.*

One cycle is required to compress a line-based sub-block, whereas it takes four cycles to decompress one 8×4 block. To read a 1080p@60fps video sequence, the required throughput of the decompressor is 186.6 Msamples/s under 4:2:0 sampling. The throughput of the decompress core can

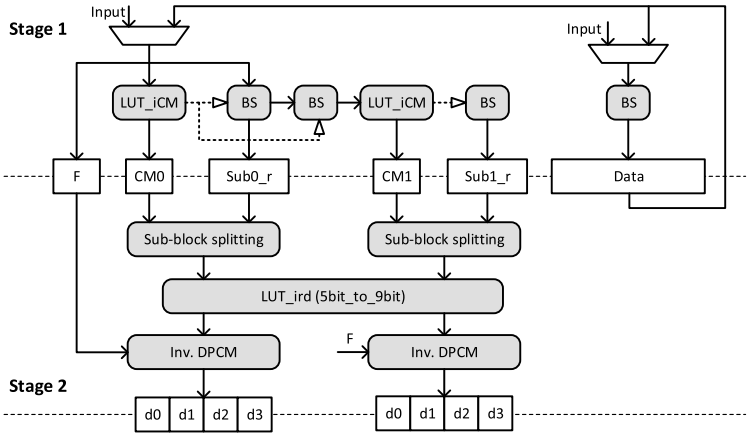

FIGURE 12. Two-stage pipelined decompress core architecture, where "BS" denotes "barrel shifter".

reach up to 2.4 Gsamples/s at 300 MHz, that is, 12 times more than the requirement of real-time 1080HD video. Hence, the proposed decompress core can support a 12-scale detection core even without multi-scale optimization, such as [6].

V. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed lossy IIC framework, we test its compression performance on the VOC2007 [23] and ImageNet ILSVRC-2012 [24] validation datasets. Because the image quality decreases because of lossy compression, the detection performance of the restored images is tested. Among the best available vision algorithms, we select two widely used algorithms for the two tasks of detection and classification. One is HOG-based deformable part models (DPM) [25] for detection. The other algorithm is CNN with the model of AlexNet. Moreover, another detection algorithm of histogram of sparse codes [28] is also tested to verify the feasibility of the non-gradient-based hand-crafted feature. A tradeoff between compression and vision performance is then presented. Finally, the hardware implementation results are presented together with the analysis of power consumption.

A. COMPRESSION PERFORMANCE

The required memory traffic for writing or reading input images is proportional to the data size of compressed images. Hence, the data reduction ratio (DRR) is used to evaluate compression performance, which is the percentage of the reduced data size compared with the original data size.

$$DRR = (1 - \frac{Compressed\ data\ size}{Original\ data\ size}) \times 100\% \qquad (1)$$

All 4,952 images from 20 object categories for the VOC2007 test are used. Because of the large size of ImageNet, we test the first 10,000 images from its validation dataset. Moreover, the compression performance of the lossless SBT coding [16] is also simulated for comparison.

Table 5 shows the compression performance of the lossless SBT and proposed lossy GOQ coding based on QC shown in Table 1. The DRRs of the proposed lossy IIC are 49.63% and 50.56% for the VOC2007 test and ImageNet2012 validation dataset, respectively. Compared with lossless SBT compression, the proposed lossy GOQ coding further improves the DRR by 28.7% and 29.2% on average for VOC2007 and ImageNet, respectively.

TABLE 5. Average DRR, CR, and PSNR of the pixel map.

	VOC2007		ImageNet2012	
	SBT [16]	GOQ	SBT [16]	GOQ
DRR (%)	28.73	49.63	30.17	50.56
Pixel PSNR (dB)	-	49.63	-	50.18

B. DETECTION PERFORMANCE

To estimate the influence of image quality loss on the detection performance of computer vision algorithms, the restored images are further detected. For the HOG-based DPM detection, we use the model trained on the VOC2007 dataset and provided by [26]. The detailed detection average precisions (APs) are shown in Table 6. Compared with the original image, the mean APs (MAPs) of the proposed lossy GOQ coding decreases by 0.135% without context optimization. The MAPs are slightly improved by the restored lossy images with context optimization [26]. Detection with the restored lossy images results in almost no loss in MAPs.

C. TRADEOFF BETWEEN COMPRESSION AND VISION PERFORMANCE

The performance of compression and vision is influenced by the QC in lossy GOQ coding (described in Section III.A). With a large quantization value, memory traffic and power dissipation are less, but vision accuracy decreases considerably. Therefore, a tradeoff between compression and vision

TABLE 6. Detection average precision (AP, %) for the VOC2007 test using the HOG-based deformable part model [26].

Class	Original [26]		Proposed GOQ	
	w/o [a]	w/ [a]	w/o [a]	w/ [a]
Aeroplane	33.2	33.2	32.9	33.9
Bicycle	59.3	60.8	59.1	61.2
Bird	10.3	10.2	10.3	10.3
Boat	15.7	16.1	15.9	16.1
Bottle	26.6	27.3	25.8	26.3
Bus	51.3	54.1	50.3	52.5
Car	53.8	58.1	53.9	58.4
Cat	22.5	23.0	22.5	23.0
Chair	20.1	20.0	20.2	20.5
Cow	24.3	24.2	24.3	24.6
Diningtable	26.9	26.8	26.3	26.3
Dog	12.6	12.7	12.5	12.4
Horse	56.5	58.1	57.2	58.1
Motorbike	48.5	48.2	48.1	48.7
Person	43.2	43.2	42.8	43.0
Pottedplant	13.4	12.0	13.6	13.3
Sheep	20.7	20.9	21.1	21.6
Sofa	35.8	35.8	35.3	35.9
Train	45.2	46.0	45.2	45.9
Tvmonitor	42.0	43.4	41.8	42.8
mean (MAP [b])	33.095	33.705	32.960	33.745
ΔMAP [c]	-	-	-0.135	0.040

[a] *w/o: without context, w/: with context [26].*
[b] *MAP is the mean of AP over classes [23] [33] [34], while AP is the area under the precision/recall curve of detection for per-class.*
[c] $\Delta MAP = MAP_{GOQ} - MAP_{ori}$.

performance is presented based on the experimental results of widely applied HOG-based DPM [26], Caffe [27] and histogram of sparse codes [28]. For the battery-less case, the acceptable detection/classification accuracy drop can be 1-10% to save more power, whereas it is less than 0.1-1% for the full-battery case.

1) HOG-BASED DPM ON VOC2007

According to the QC in Table 7, all the test images from VOC2007 are compressed and decompressed by the proposed lossy GOQ coding. As a comparison, the linear quantization

TABLE 7. Quantization coefficients (QC) for each lossy compression version ($v_0 - v_2$) and coding mode (CM).

CM	v_0	v_1	v_2
[A,B,C,D]	[1,3,3,5,5]	[2,5,7,9,11]	[3,7,9,11,11]
E	[5,5,5,5]	[13,13,15,15]	[13,13,15,15]
G	[7,7,7,7, 7,7,7,7]	[19,21,21,21, 21,21,21,21]	[15,19,21,21, 21,21,21,21]

that involves quantizing residuals in lossless DPCM-based SBT by 2^n (SBT_nbit) is estimated. Furthermore, the proposed IIC is compared with four previous lossless/lossy IICs [16], [17], [19] and directly discarding the trailing n bits of the original images (divided by 2^n, Image_nbit).

The relationship between the compression performance of the DRR and detection performance of the MAP are shown in Fig. 13. Compared with linear quantization (SBT_nbit) and previous lossy works (Song *et al.* [17] and Gupte *et al.* [19]), the detection accuracy of the proposed GOQ IIC is always better. Although the lossless IIC [18] could achieve a DRR of 50.24% without a decrease in accuracy, its high complexity and data dependency make it difficult to support high-throughput DRAM access. To obtain a similar DRR, the MAP degradation of the proposed IIC is less than 0.135%.

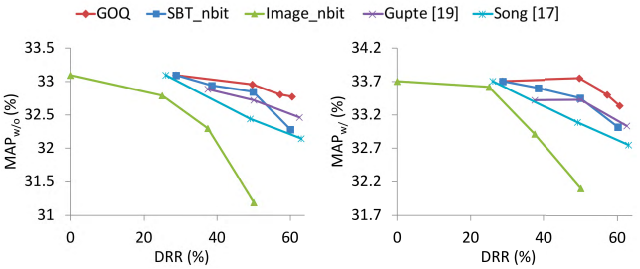

FIGURE 13. Comparison results of the DRR and MAP (tested in DPM).

The PSNR is widely used as the cost function to efficiently evaluate image quality based on a visual experience. However, it is not very appropriate for computer vision applications that are based on detection accuracy. With similar PSNR values (a difference of 0.0056dB), the difference in the MAP can be up to 0.79%. This difference is even larger than the decrease in the MAP caused by lossy GOQ compression. Therefore, detection performance cannot be estimated directly from the PSNR values of the input images.

2) AlexNet on ImageNet2012

In CNN-based classification, the AlexNet Caffe model trained on ImageNet2012 and provided by [27] is used in our experiments. Because a fixed-point arithmetic operation is used in some real ASIC CNN implementations, such as [30], we evaluate the classification accuracies of the top-1 and top-5 based on a 16-bit fixed-point weight model. The results are shown in Fig. 14. When the DRR ranges from 50.56% to 61.22%, the corresponding top-1 accuracies decrease by 0.28%–0.61%, whereas there is almost no loss in the top five accuracies. For standard floating-point weights, the top one accuracy is 57.3%, which is 4.4% greater than the 16-bit fixed-point case. Compared with the accuracy degradation from fixed-point processing, the effect of lossy IIC on input images is much lower. Compared with previous works, the proposed GOQ achieves slightly better top-1 and top-5 accuracies. However, this improvement is not as large as that for gradient-based feature extraction, such as HOG.

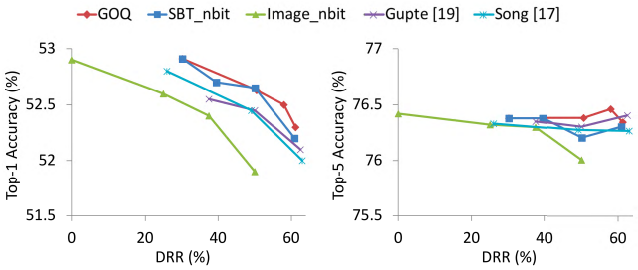

FIGURE 14. DRR and classification accuracy of the top-1 and top-5 on a 16-bit fixed-point AlexNet weight model. (Fractional length is 8 bits. Without using the lossy compression, the top-1 and top-5 accuracies of 16-bit fixed-point AlexNet model are 52.9 % and 76.4%, respectively.)

3) HISTOGRAM OF SPARSE CODES ON VOC2007

Additionally to HOG-based DPM in Section V.C.1, we also test another hand-crafted feature of histogram of sparse codes [28], that is, the non-gradient. For the VOC2007 dataset, the pretrained part model in [28] is used and the MAP results for different DRRs are shown in Fig. 15. With lossy IICs, there is almost no degradation in the MAP. Even if the proposed GOQ is not always better than the linear quantization (SBT_nbit), the maximum difference for similar DRRs is only 0.125%, which is a slight fluctuation. Thus, although the proposed GOQ is aimed at preserving gradients, it is still as efficient for vision algorithms with non-gradient features.

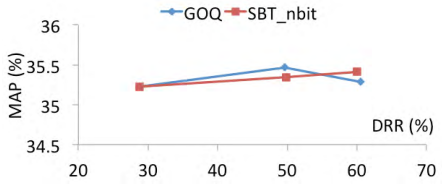

FIGURE 15. Comparison result of GOQ and linear quantization (SBT_nbit), which is tested in histogram of sparse codes [28].

D. RESULTS OF THE HARDWARE IMPLEMENTATION

1) PROCESSING FORMAT CONVERSION

The line buffer is a straightforward and widely applied solution to convert the scanning order of an image. To obtain a $k \times k$ block from a line-based scanning image, the corresponding k lines of the image should be stored on-chip. Thus, we compare the hardware cost of the proposed line-to-block conversion with a line buffer. Compared with the overall on-chip memory, a proportion of the input line buffer is related to input image resolution and processing scales. It is larger for higher resolution and more scales. For example, in the 12-scale HOG detector [6] for 1080HD video, although several optimizations have been proposed to reduce the size of the input line buffer, the latter still occupies 25.3% of the total on-chip memory. Even if the input line buffer is not the dominant fraction, saving it with a sufficiently small overhead is still valuable.

Table 8 shows the comparison of the required buffer size. The ratio is calculated as the buffer size of the line buffer divided by that of the proposed conversion. It becomes larger as the resolution increases from low to high. Therefore, the proposed conversion is suitable for high-resolution images that are difficult for the line buffer.

TABLE 8. Required buffer size during conversion (bytes).

Vision block	Resolution	Line buffer	Proposed	Ratio
8×8	C	$8.25C$	$1/4C + 896$	-
	720p	5,940	1,076	x5.5
	1920p	15,840	1,376	x11.5
11×11	C	$11.25C$	$5/4C+1,298$	-
	720p	8,100	2,198	x3.7
	1920p	21,600	3,698	x5.8

Note: C is the resolution of the image in width. In this example, w = 256-bit, m = 8, n = 4, and the vision block size is set to 8×8 and 11×11.

2) LOSSY INPUT IMAGE COMPRESSION

Table 9 shows the hardware implementation results of the proposed IIC core. The compress core achieves a throughput of 800 Mpixels/s at 300 MHz under a 4:2:0 video sampling, and the decompresser is designed with a throughput of 1.6 Gpixels/s. This throughput can support up to 12-scale 1080p@60fps video for real-time detection, which is sufficient to support all previous vision cores.

3) ANALYSIS OF RESOURCE CONSUMPTION

The IIC framework reduces power dissipation by reducing the amount of external memory traffic; however, it also consumes power as an additional portion to the vision core. The power presented in Table 9 is estimated with the Synopsys Design Compiler from the switching activity statistics from the post-synthesis simulation. For the proposed IIC core, the total power dissipation is the sum of memory traffic and the IIC core.

When images are read with the IIC core, power is consumed by memory traffic and the decompress core. According to DRRs of different quantization versions depicted in Fig. 13, Fig. 16 shows an example of the total power dissipation for reading a 1080p@60fps video. Under 4:2:0 sampling, the required throughput for reading 1080p video is 186.6 MB/s. Without optimization, the power consumption is 96.6 mW (186.6 MB/s×517.63 mW/GB/s [31]). With the proposed IIC, the required power for reading can be reduced by 46.9%–57.8%. The overhead of decompress core is approximately 5% compared with the power reduction by the IIC core.

Based on an HOG-based DPM detector [32], the analysis of power dissipation in a system is presented in Table 10. Because the detector is implemented on older 65 nm CMOS technology, the DDR2 interface is considered as matching technology for DRAM. To read a 1080p@30fps video,

TABLE 9. Comparison of hardware implementations.

	Kim [16]	Gupte [19]	Lian [18]		This Work	
	comp. / decomp. [a]	comp.	comp.	decomp.	comp.	decomp.
CMOS Tech. (nm)	180	65	65		40	
Frequency (MHz)	180	250	578	599	300	
Latency (cycles)	N/A	N/A	N/A		2 [b]	
Gate Count (k)	36.1	14	36.5	34.7	4.95	9.75
Throughput (samples/cycle)	5.1 / 14.2	2.5 [c]	2.67	1.33	4	8
Throughput (Gsamples/s)	0.9 / 2.6	0.625 [c]	1.54	0.78	1.2	2.4
Power (mW)	-	1.35.	5.3	5.0	2.05	2.60
Norm. Power [d] (pJ/samples)	-	2.16	3.44	6.41	1.70	1.08

[a] *comp. / decomp.: compressor and decompressor cannot be used simultaneously.*
[b] *The latency for compressing or decompressing a sub-block ($n\times1$). To decompress an $n\times k$ block, the latency is $k/2+1$ cycles.*
[c] *Values are calculated from the given bandwidth in [19] TABLE II.*
[d] *Norm. Power = Power / Throughput.*

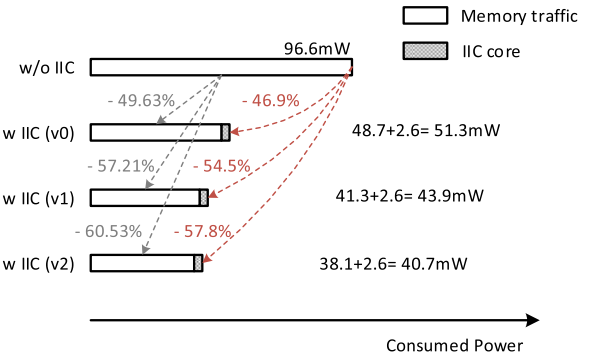

FIGURE 16. Simulated result of the consumed power related to memory traffic. (DDR3-1333 interface [35]. For v0-v2, only the quantization coefficients are different as shown in Table 7; thus the power of the IIC core is the same.)

TABLE 10. Analysis of power dissipation for the detection based on DPM.

Power (mW)		w/o IIC	w/ IIC	reduced percent
Memory traffic	write	90.7	40.9	55.0%
	read	90.7	41.4	54.3%
Detection core [32]		58.6	58.6	0%
Total (detection process)		149.3	100.0	33.0%
Total (entire system)		240.0	140.9	41.3%

Note: 1080p@30fps, DDR2-667 [31] [36]. For the proposed IIC, GOQ is set to v_1 in Table 7, DRR = 57.21%.

90.7 mW (93.3 MB/s×971.51 mW/GB/s [31]) is consumed. For the detection process that includes a decompress core and detector, 33.0% of power can be reduced by the proposed lossy IIC. For the entire system, such as that in Fig. 2, which contains processes for both reading and writing images, 41.3% of the total power dissipation is saved.

Using the proposed IIC framework, although the system-level power dissipation can be reduced, the total area (i.e. gate count) increases because of the additional compression. However, as shown in Table 11, the increase is less than 3% of detection systems. Compared with the saved power of 41.3%, the increased gate count is acceptable.

TABLE 11. Analysis of consumed gates.

	w/ IIC	object detector	
		HOG+SVM [6]	HOG+DPM[32]
Logic gates (k)	14.7	490∼498	3283
Increased gates[a]	-	3.0%	0.45%

[a] *the increased percentage of logic gates in object detection systems using the proposed IIC.*

In addition to the above analysis of power dissipation in the detection process, we also provide a similar analysis of the classification of AlexNet. In the CNN accelerator [37], AlexNet consumes 450 mW on 65 nm CMOS technology. Hence, the DDR2 interface is considered as the matching technology for DRAM, and 45 images are read from DRAM per second. The power consumption for the classification process is shown in Table 12, which includes the power of the classification core and reading input images from DRAM. The resolution of input images for AlexNet is 227×227. However, for a system on chip, input images are typically shared by multiple components, such as the object detector, image classifier, and video encoder; hence, images from the

TABLE 12. Analysis of power dissipation for the classification with AlexNet.

resolution	power (classification process, mW)		reduced percent
	w/o IIC	w/ IIC	
80×60	450.6	452.9	-0.5%
227×227	456.8	455.5	0.3%
720 ×1280	570.9	504.3	11.7%
1080×1920	722.0	569.0	21.2%
4288×2848	2051.7	1138.0	44.5%

Note: 45 fps, DDR2-667 [31] [36]. For the proposed IIC, GOQ is set to v_1 in Table 7, DRR = 57.21%.

sensor are stored directly in DRAM without resizing for a certain component. For the classification process, the resolution of input images depends on the sensor, and five typical resolutions are considered in Table 12, including the minimum/maximum resolution in ImageNet (80×60, 4288×2848), input resolution of AlexNet (227×227), and two typical resolutions of sensor (720p, 1080p). Except for the minimum resolution of 80×60, the other resolutions can achieve 0.3–44.5% power reduction for the classification process. For the entire system, such as that shown in Fig. 2, the power reduction with the proposed IIC is 0.7–50%.

VI. CONCLUSION

To reduce power dissipation in embedded systems for computer vision, we present a lossy compression framework. By compressing the input images, memory traffic from the external DRAM substantially decreases, thereby reducing power consumption. Because of the application of lossy compression, a tradeoff between the detection/classification accuracy and compression performance is explored according to experimental results on the widely accepted HOG-based DPM, AlexNet and histogram of sparse codes.

In our future studies, this framework will be extended to improve energy efficiency for accessing feature maps and weights. Moreover, exploring an efficient input image compression algorithm remains vital work, such as compression in the frequency domain. Because of the complexity of the transformation, a tradeoff between the overhead caused by compression and energy consumption may be involved.

APPENDIX

For clarity, abbreviations that are frequently used in this paper are summarized in Table 13.

TABLE 13. Summary of abbreviations.

Abbreviation	Full Name (used or defined in page)
IIC	Input Image Compression (pp. 2)
GOQ	Gradient Oriented Quantization (pp. 3)
QC	Quantization Coefficients (pp. 5)
CM	Coding Mode (pp. 4)
DPCM	Differential Pulse-Code Modulation [14] (pp. 3)
SBT	Significant Bit Truncation [16] (pp. 4)
CNN	Convolutional Neural Network (pp. 1)
HOG	Histogram of Oriented Gradients [8] (pp. 1)
SIFT	Scale-Invariant Feature Transform (pp. 2)
SVM	Support Vector Machine (pp. 1)
DPM	Deformable Part Models [25] (pp. 8)
DRR	Data Reduction Ratio (pp. 8)
PSNR	Peak Signal-to-Noise Ratio (pp. 2)
AP	Average Precision (pp. 9)
MAP	Mean of Average Precision [23] [33] [34] (pp. 9)

Note: The above three sections are related to input image compression, computer vision and the measurements for evaluating performance, respectively.

REFERENCES

[1] V. Haltakov, H. Belzner, and S. Ilic, "Scene understanding from a moving camera for object detection and free space estimation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 105–110.
[2] M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications," in *Proc. 2nd Int. Workshop Networked Sens. Syst.*, 2005, pp. 1–6.
[3] R. Nallusamy and K. Duraiswamy, "Solar powered wireless sensor networks for environmental applications with energy efficient routing concepts: A review," *Inf. Technol. J.*, vol. 10, no. 1, pp. 1–10, Jan. 2011.
[4] M. Hahnle, F. Saxen, M. Hisung, U. Brunsmann, and K. Doll, "FPGA-based real-time pedestrian detection on high-resolution images," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 629–635.
[5] F.-C. Huang, S.-Y. Huang, J.-W. Ker, and Y.-C. Chen, "High-performance SIFT hardware accelerator for real-time image feature extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 340–351, Mar. 2012.
[6] A. Suleiman and V. Sze, "An energy-efficient hardware implementation of HOG-based object detection at 1080HD 60 fps with multi-scale support," *J. Signal Process. Syst.*, vol. 84, no. 3, pp. 325–337, Sep. 2015.
[7] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, and E. Culurciello, "Hardware accelerated convolutional neural networks for synthetic vision systems," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, May/Jun. 2010, pp. 257–260.
[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[9] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. Int. Conf. Field Program. Logic Appl. (FPGA)*, 2015, pp. 161–170.
[10] Y. Chen *et al.*, "DaDianNao: A machine-learning supercomputer," in *Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2014, pp. 609–622.
[11] Z. Wang, H. Xiao, W. He, F. Wen, and K. Yuan, "Real-time SIFT-based object recognition system," in *Proc. Int. Conf. Mechatronics Automat. (ICMA)*, 2013, pp. 1361–1366.
[12] M. Peemen, A. A. A. Setio, B. Mesman, and H. Corporaal, "Memory-centric accelerator design for convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 13–19.
[13] D. Zhou *et al.*, "A 4 Gpixel/s 8/10b H.265/HEVC video decoder chip for 8K ultra HD applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan./Feb. 2016, pp. 266–268.
[14] D. Zhou *et al.*, "A 530 Mpixels/s 4096×2160@60fps H.264/AVC high profile video decoder chip," *IEEE J. Solid-State Circuits*, vol. 6, no. 4, pp. 777–788, Apr. 2011.
[15] L. Guo, D. Zhou, and S. Goto, "A new reference frame recompression algorithm and its VLSI architecture for UHDTV video codec," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2323–2332, Dec. 2014.
[16] J. Kim and C.-M. Kyung, "A lossless embedded compression using significant bit truncation for HD video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 848–860, Jun. 2010.
[17] L. Song, D. Zhou, X. Jin, and S. Goto, "A constant rate bandwidth reduction architecture with adaptive compression mode decision for video decoding," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2010, pp. 2017–2021.
[18] X. Lian, Z. Liu, W. Zhou, and Z. Duan, "Lossless frame memory compression using pixel-grain prediction and dynamic order entropy coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 223–235, Jan. 2016.
[19] A. D. Gupte, B. Amrutur, M. M. Mehendale, A. V. Rao, and M. Budagavi, "Memory bandwidth and power reduction using lossy reference frame compression in video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 225–230, Feb. 2011.
[20] T. L. B. Yng, B. G. Lee, and H. Yoo, "A low complexity and lossless frame memory compression for display devices," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 1453–1458, Aug. 2008.
[21] S. Kim, D. Lee, J.-S. Kim, and H.-J. Lee, "A high-throughput hardware design of a one-dimensional SPIHT algorithm," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 392–404, Mar. 2016.
[22] L. Guo, D. Zhou, J. Zhou, and S. Kimura, "Embedded frame compression for energy-efficient computer vision systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–4.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[24] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and F.-F. Li. (2012). *ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012)*. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2012/

[25] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[26] R. Girshick, P. Felzenszwalb, and D. Mcallester. (2012). *Discriminatively Trained Deformable Part Models, Release 5*. [Online]. Available: http://people.cs.uchicago.edu/~rbg/latent-release5

[27] Y. Jia *et al.* (Jun. 2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: https://arxiv.org/abs/1408.5093

[28] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3246–3253.

[29] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3306–3313.

[30] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan./Feb. 2016, pp. 262–264.

[31] J. T. Pawlowski, "Hybrid memory cube: Breakthrough DRAM performance with a fundamentally re-architected DRAM subsystem," in *Proc. 23rd Hot Chips Symp.*, 2011, pp. 1–24.

[32] A. Suleiman, Z. Zhang, and V. Sze, "A 58.6 mW real-time programmable object detector with multi-scale multi-object support using deformable parts model on 1920×1080 video at 30 fps," in *Proc. Symp. VLSI Technol. Circuits (VLSI)*, 2016, pp. 184–185.

[33] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[34] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[35] *DDR3 SDRAM Standard*, Standard JESD79-3F, JEDEC, Jul. 2012.

[36] *DDR2 SDRAM Specification*, Standard JESD79-2E, JEDEC, Mar. 2009.

[37] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

LI GUO received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, the M.E. degree from Waseda University, Kitakyushu, Japan, in 2013, and the M.E. degree from Shanghai Jiao Tong University in 2015. She is currently pursuing the Ph.D. degree with Waseda University. Her current research interests include algorithms and VLSI architectures for multimedia signal processing and neural networks. She is being supported by the Ministry of Education, Culture, Sports, Science and Technology, Tokyo, Japan.

DAJIANG ZHOU received the B.E. and M.E. degrees from Shanghai Jiao Tong University, China, and the Ph.D. degree in engineering from Waseda University, Japan, in 2010. He is currently with Waseda University as an Assistant Professor with the Graduate School of Information, Production and Systems. His interests are in algorithms and implementations for multimedia and communications signal processing, especially in low-power high-performance VLSI architectures for video codecs, including H.265/HEVC and H.264/AVC.

Dr. Zhou was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science from 2009 to 2011. He received a number of awards, including the Best Student Paper Award of VLSI Circuits Symposium 2010, the International Low Power Design Contest Award of ACM ISLPED 2010, the 2013 Kenjiro Takayanagi Young Researcher Award, and the Chinese Government Award for Excellent Students Abroad of 2010. His work on an 8K UHDTV video decoder VLSI chip was granted the 2012 Semiconductor of the Year Award of Japan.

JINJIA ZHOU (S'12–M'13) received the B.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2007, and the M.E. and Ph.D. degrees from Waseda University, Kitakyushu, Japan, in 2010 and 2013, respectively. She was a Researcher with Waseda University from 2013 to 2016. She is currently an Associate Professor with Hosei University, Tokyo, Japan. Her current research interests include algorithms and VLSI architectures for video coding. He received the Research Fellowship of the Japan Society for the Promotion of Science from 2010 to 2013. She was a recipient of the Chinese Government Award for Excellent Students Abroad of 2012.

SHINJI KIMURA received the B.E., M.E., and Dr.Eng. degrees in information science from Kyoto University, Kyoto, Japan, in 1982, 1984, and 1989, respectively. He has been an Assistant Professor with Kobe University since 1985, has been an Associate Professor with the Nara Institute of Science and Technology since 1993, and has been a Professor with Waseda University since 2002. He was a Visiting Scientist with Carnegie Mellon University from 1989 to 1990 and was a Visiting Scholar with Stanford University from 2000 to 2001.

He is interested in the formal and timing verification of logic circuits, the hardware/software co-design methodologies, reconfigurable hardware, and the low-power design. He is a member of the Information Processing Society of Japan and the IEEE Computer Society. He has served an Executive Committee Member of ICCAD 2011 and 2012 and a General Chair of ASP-DAC 2013. He has been a fellow of IEICE since 2015.

SATOSHI GOTO (S'69–M'77–SM'84–F'86–LF'11) received the B.E. and M.E. degrees in electronics and communication engineering and the Doctor of Engineering degree from Waseda University in 1968, 1970, and 1981, respectively. He joined NEC Laboratories in 1970, where he was involved in LSI design, multimedia systems, and software as a GM and a VP. Since 2003, he has been a Professor with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan, where he is currently a Professor Emeritus. His main interests are in VLSI design methodologies for multimedia and mobile applications.

He has published seven books and over 300 technical papers in international journals and conferences. He was a Board Member of the IEEE CAS Society. He is a fellow of IEICE and a member of the Science Council of Japan. He received a number of awards and honors, including the Distinguished Achievement Awards from IEICE and the Jubilee Medal from the IEEE. He served as a GC of ICCAD and ASPDAC.