14.6 A 1.42TOPS/W Deep Convolutional Neural Network Recognition Processor for Intelligent IoE Systems

Jaehyeong Sim, Jun-Seok Park, Minhye Kim, Dongmyung Bae, Yeongjae Choi, Lee-Sup Kim

KAIST, Daejeon, Korea

Transmitting massive amounts of image and audio data acquired by Internet-of-Everything (IoE) devices to data center servers for intelligent recognition processes is impractical for energy reasons, requiring in-situ processing of such data. However, algorithms accelerated by previous recognition processors [1, 2] are limited to specific applications, therefore, each IoE device may require an application-specific accelerator. On the other hand, deep convolutional neural networks (CNNs) [3] are a promising machine-learning approach, showing state-of-the-art recognition accuracy in a wide variety of applications, including both image and audio recognition. This makes CNNs a suitable candidate for a universal recognition platform for IoE devices, as described in Fig. 14.6.1. Due to the computational complexity and significant memory requirements of CNNs, a microcontroller unit (MCU) typically used for IoE devices is incapable of producing a meaningful recognition result in an energy-efficient way. Hence, the implementation of an energy-efficient CNN processor is desired to realize intelligent IoE systems.

In this paper, we present an energy-efficient CNN processor with 4 key features: (1) a CNN-optimized neuron processing engine (NPE), (2) a dual-range multiply-accumulate (DRMAC) block for low-power convolution operations, (3) an on-chip memory architecture and a utilization scheme for reducing off-chip memory accesses, (4) kernel data compression for further reducing off-chip memory accesses.

Figure 14.6.2 shows the overall architecture of the CNN processor. 4 homogeneous CNN cores are integrated and connected by a shared memory bus. Each core contains 2 NPEs, 2 image buffers, 2 output buffers, and a kernel buffer. A single front-end/control unit fetches and decodes instructions and generates control signals for both NPEs, meaning that two NPEs operate in lockstep. Each NPE accesses a different image and output buffer, while a single kernel buffer is shared by 2 NPEs. An NPE consists of the minimum number of hardware blocks required for a single recognition process in a CNN, including 32 DRMAC blocks, 32 rectified-linear (ReLU) blocks, and 8 maxpool blocks. For the computation of a convolution layer in a CNN, an NPE executes 32 MAC blocks to perform 32 different convolution operations in parallel. A group of 8 DRMAC blocks shares a single kernel word to avoid redundant memory accesses for the same kernel word when an input image is swept with a kernel, generating 8 words for a single output image. 4 DRMAC groups are fed with the same set of 8 input image words, but with 4 different kernel words to prevent redundant memory accesses for the same input word when an input image is filtered with different kernels. In this way, the input image and kernel data provided in each cycle are maximally utilized. Finally, when convolution operations for the 32 output words are finished, the NPE runs 32 ReLU blocks to rectify the output words. For a max-pooling layer, the NPE executes 8 max-pooling blocks in parallel, and each max-pooling block selects an output word with the maximum value among 4 output words.

Most of the operations required for a single recognition process in a CNN are MAC operations. In order to reduce the power consumed by the massive number of MAC operations, a DRMAC block is implemented as shown in Fig. 14.6.3. Basically, a 24b (16.8 format) fixed-point truncated multiplier and an adder are included in a single MAC block to cover the dynamic range and precision of operands used in our target CNN benchmarks: MNIST, CIFAR-10, GTSRB, and AlexNet. This provides similar recognition accuracy to a 64b double-precision floating-point format that requires much more complex hardware. However, further analysis on the MAC operands used in CNN reveals that about 99% of operands require at most 8b for the integer part, while only 0.01% of operands need full 16b for the integer part. Additionally, operands which require 16b for the integer part are generated by repetitive accumulations of operands with small values. Therefore, instead of executing full 24b MAC operations all the time, 16b (8.8 format) MAC operations are performed at first, saving unnecessary power consumption. The 16b truncated MAC operation is implemented by masking the upper 8 bits of inputs of a 24b MAC block, which results in the elimination of the switching activity of the upper part. When an overflow flag is detected by a 16b MAC operation, the pipeline is stalled for one cycle to prevent the accumulator register from updating the wrong MAC result, and then the DRMAC block begins to operate in a full 24b mode. As a result, the DRMAC block reduces the power consumption by 56% compared to the 24b MAC block.

Transferring a large amount of image and kernel data from off-chip memory to on-chip buffers and writing intermediate image values to off-chip memory incurs a significant loss in both performance and energy-efficiency. Thus, the data of on-chip buffers are maximally reused to reduce unnecessary off-chip memory accesses, as shown in Fig. 14.6.4. First, convolution operations are performed in a tiled-manner with the size of a tile equal to the capacity of an on-chip image buffer when it is smaller than the size of an input image. That is, instead of filtering the entire image with a single kernel, only a tile is swept with as many kernels as possible. Thereafter, the data of the tile is never accessed. Second, when the tile operation is finished, on-chip input and output image buffers switch their role to the opposite one for the subsequent CNN layer since the output images of the current layer become the input images of the next layer. As a result, intermediate output images do not need to be written to off-chip memory.

To further reduce off-chip kernel data accesses, an algorithm-level modification in accord with the CNN hardware architecture is presented, as shown in Fig.14.6.5. Based on the observation that a group of kernels in a CNN exhibits a high correlation between them, principal component analysis (PCA) is carried out on the kernel data, and fewer basic kernels are extracted. Then, the original group of kernels is generated by weighted sums of such basic kernels. In other words, only basic kernels are saved in on-chip buffers and constant values needed for the weighted sum are transferred from off-chip memory, reducing the overhead of transferring the whole group of kernels at a little cost of on-chip kernel generation process and recognition accuracy loss. For the kernel generation process, since weighted sum operations are equal to a series of MAC operations, the existing DRMAC blocks are used without any extra hardware overhead. This on-chip kernel generation technique is applied for target CNN benchmarks. Off-chip kernel data are reduced by the maximum 92%, with only 0.68% recognition accuracy loss.

Figure 14.6.6 shows chip features, and shows a comparison with previous CNN processors. The die micrograph is shown in Fig. 14.6.7. The CNN processor is fabricated in 65nm 1P8M CMOS technology. It integrates 3.2M logic gates, including 36KB of on-chip memory within a 4×4mm^2 die. The maximum supported kernel size is 15×15. The power consumption is 45mW, operating at 125MHz clock frequency with a 1.2V supply voltage, measured by performing convolution on the data extracted from the convolution layers of the MNIST benchmark. We achieve 1.42TOPS/W energy efficiency, represent 5.59×, 4.06×, and 1.52× improvements over previous CNN processors [4-6], which makes it a promising universal recognition platform for intelligent IoE systems.

Acknowledgements:
Chip fabrication was supported by IDEC at Korea Advanced Institute of Science and Technology (KAIST), Korea. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A05004316, No. 2010-0028680)

References:
[1] Y. Kim et al., "A 0.5V 54mW Ultra-Low-Power Recognition Processor with 93.5% Accuracy Geometric Vocabulary Tree and 47.5% Database Compression," *ISSCC Dig. Tech. Papers*, pp. 330-331, 2015.
[2] M. Price et al., "A 6mW 5K-Word real-time speech recognizer using WFST models," *ISSCC Dig. Tech. Papers*, pp. 454-455, 2014.
[3] Y. LeCun et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
[4] P. Pham et al., "NeuFlow: Dataflow Vision Processing System-on-a-Chip," *IEEE Midwest Symp. on Circuits and Systems*, pp. 1044-1047, 2012.
[5] Y. Chen et al., "DaDianNao: A Machine-Learning Supercomputer." *IEEE/ACM Int. Symp. on Microarchitecture*, pp. 609-622, 2014.
[6] T. Chen et al., "A High-Throughput Neural Network Accelerator," *IEEE Micro*, vol. 35, no. 3, pp. 24-32, 2015.

 978-1-4673-9467-3/16/$31.00

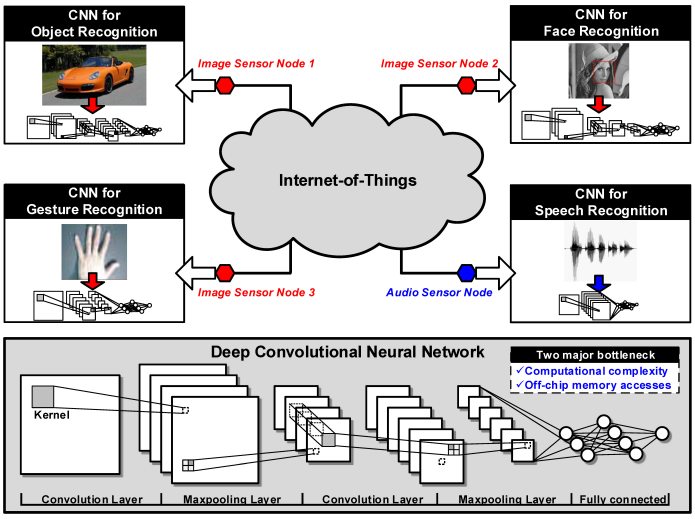

Figure 14.6.1: CNN and its application to an IoE system.

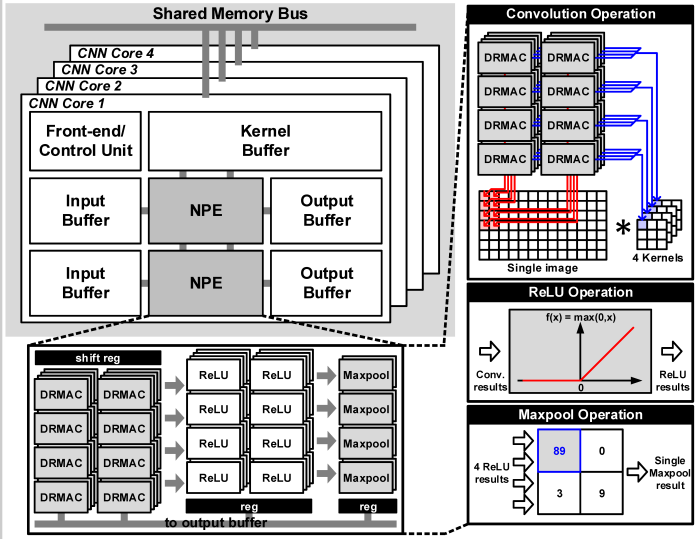

Figure 14.6.2: Overall architecture.

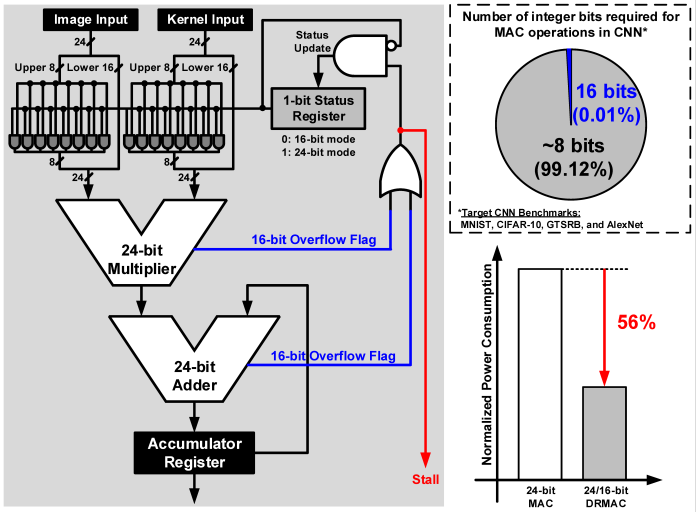

Figure 14.6.3: Dual-range multiply-accumulate block.

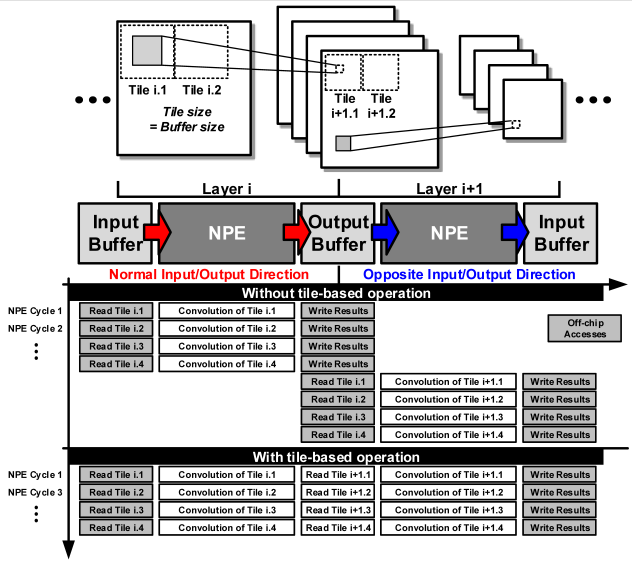

Figure 14.6.4: Tile-based operation.

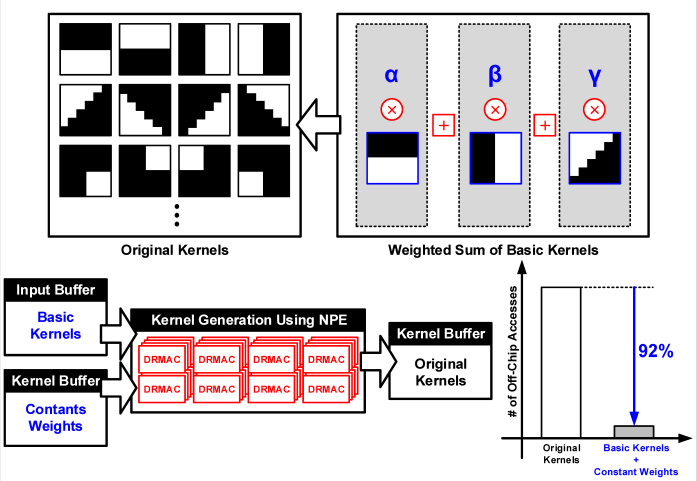

Figure 14.6.5: On-chip kernel generation technique.

Technology	65nm CMOS (1P8M)
Power Supply	1.2V (Core) / 3.3V (IO)
Chip Size	4mm x 4mm
Operating Frequency	125MHz
Power Consumption	45mW
Gate Counts	3.2M gates
On-chip Memory	36KB
Performance	64GOPS*
Energy Efficiency	1.42TOPS/W

* 1 MAC is defined as 2OPs (ADD + MUL)

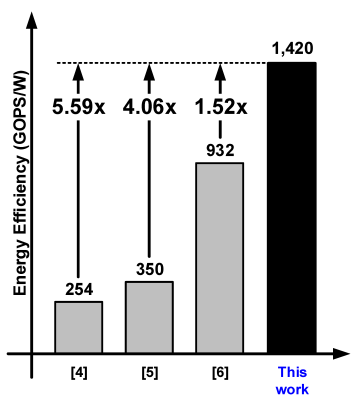

Figure 14.6.6: Chip specification and comparison.

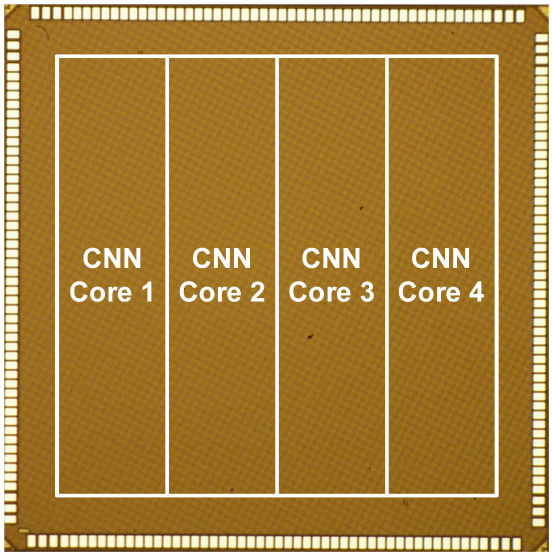

Figure 14.6.7: Chip micrograph.

978-1-4673-9467-3/16/$31.00