Journal of Imaging Science and Technology® 63(5): 050404-1–050404-4, 2019.

Area-Efficient Two-Dimensional Separable Convolution Structure

Hyeonkyu Kim and Hoyoung Yoo
Chungnam National University, Daehak-ro 99, Yuseong-gu, Daejeon, Republic of Korea
E-mail: hyyoo@cnu.ac.kr

Abstract. *In a recent image processing system, convolution operations play a significant role in manipulating image and extracting features from images. Due to the increase of kernel sizes, the image processing hardware suffers from severe hardware complexity and power consumption. In this article, an area-efficient structure is proposed for two-dimensional separable convolution operations. Since a separable convolution allows to translate a two-dimensional convolution into two one-dimensional convolutions, it is possible to compute row-wise and column-wise convolutions independently. Whereas the previous work performs such one-dimensional convolutions in sequence, the proposed structure computes the one-dimensional convolutions simultaneously by rescheduling the computational sequence. Experimental results show that the proposed structure saves approximately 80% and 38% of the hardware resources compared to the conventional and previous structures, respectively.* © *2019 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2019.63.5.050404]

1. INTRODUCTION

Since the convolution neural network (CNN) makes a breakthrough in a variety of applications [1], the convolution operation becomes one of essential operations in image processing systems [2, 3]. Depending on the convolutional filter called kernel, the convolution operations can accomplish a wide range of imaginary effects including blurring, sharpening, embossing, edge detection, and more [4]. In a recent image processing system, the kernel size becomes larger in order to achieve more accurate and reliable convolutional results. However, a large amount of computations is inevitable due to the increase of kernel size. Many approaches [5–7] have been proposed to mitigate the severe computations in two-dimensional convolutions. most researches [8, 9] focused on circuitry optimizations of arithmetic operators including multipliers and adders. Recently, Debasish Mukherjee [10] proposed a more efficient method using a separable convolution, which translates a two-dimensional convolution into two one-dimensional convolutions. Since it is possible to compute row-wise and column-wise convolutions independently, the separable convolution [10] can reduce the computational complexity by 71% compared to the conventional two-dimensional convolution. In this article, we propose an area-efficient structure for separable convolutions by rescheduling the computational sequences. Whereas the separable convolution [10] performs one-dimensional convolutions in sequence, the proposed structure computes the one-dimensional convolutions simultaneously. The rest of this article is organized as follows. Section 2 explains the background and Section 3 describes the proposed area-efficient structure. Experimental results are discussed in Section 4 and concluding remarks are made in Section 5.

Received Nov. 15, 2018; accepted for publication May 8, 2019; published online Oct. 3, 2019. Associate Editor: Zeev Zalevsky.
1062-3701/2019/63(5)/050404/4/$25.00

2. BACKGROUND

The convolution operation is the process of adding each element to its neighboring elements weighted by a kernel so as to perform manipulating image and extracting features. Since pixels are located in row and column directions, a two-dimensional convolution is more widely applied to image processing. The two-dimensional convolution can be obtained by simply expanding one-dimensional convolution in an additional orthogonal direction. More precisely, given an input pixel x_{ln} for $1 \le l \le L$ and $1 \le n \le N$, the two-dimensional convolution is calculated as

$$y_{ln} = \sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} \times x_{(l+i)(n+j)}. \tag{1}$$

Figure 1 depicts an example of the conventional two-dimensional convolution structure [5] for $K = 3$. The conventional structure [5] consists of K^2 multipliers, K^2 flip-flops, and $K^2 - 1$ adders to obtain one output pixel in a single cycle. Each multiplier computes a weighted value y_{ln}^k for $1 \le k \le K^2$, and the stored K weighted values y_{ln}^k in the flip-flops are added to generate a convolutional result y_{ln} through the adder tree.

Among various approaches to reduce severe computations in the two-dimensional convolution as shown in Eq. (1), separable two-dimensional convolution [10] is one of the most efficient algorithms, which translate one two-dimensional convolution into two one-dimensional convolutions. When the two-dimensional $K \times K$ kernel w can be decomposed into $K \times 1$ vertical kernel v and $1 \times K$ horizontal kernel h, we can also decompose Eq. (1) into two one-dimensional convolutions as

$$t_{ln} = \sum_{i=1}^{K} h_i \times x_{l(n+i)}, \quad y_{ln} = \sum_{i=1}^{K} v_i \times t_{(l+i)n}, \tag{2}$$

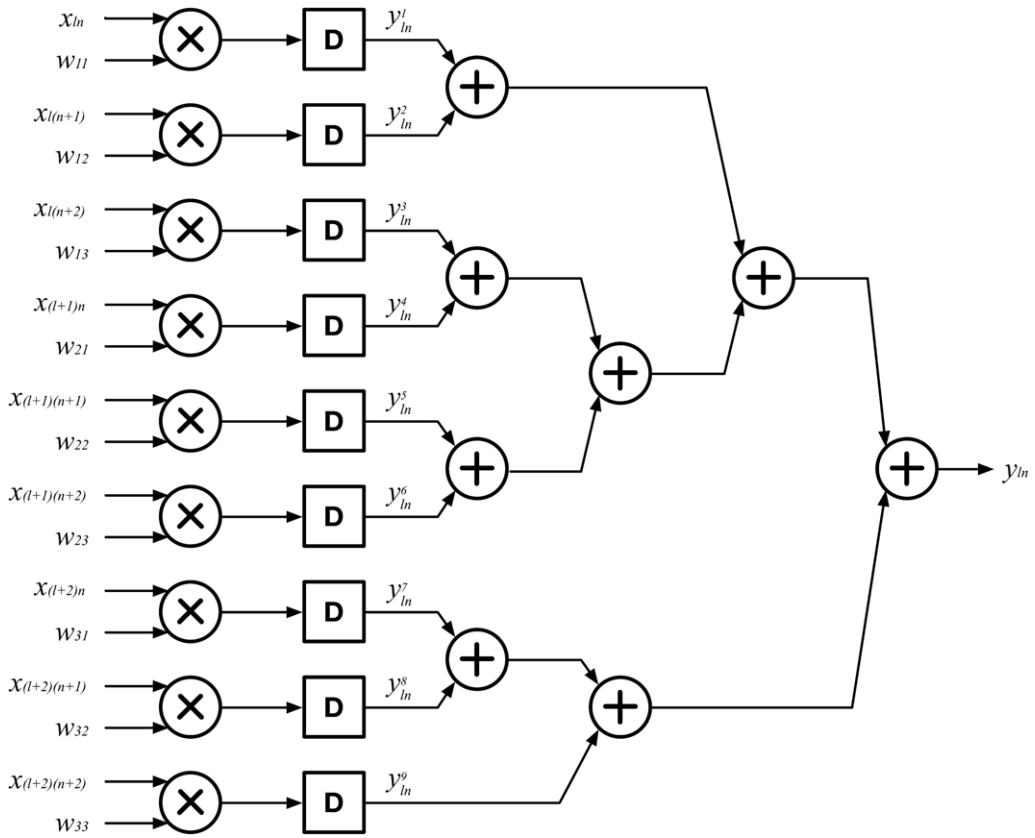

Figure 1. Structure of the conventional two-dimensional convolution for $K = 3$.

where h_i, v_i, and t_{ln} indicate an element in decomposed horizontal kernel, decomposed vertical kernel, and the result from the one-dimensional convolution between input image and horizontal kernel, respectively. Separable convolution first performs the one-dimensional horizontal convolution with h_i for $1 \le i \le K$ resulting in t_{ln}. Secondly, the one-dimensional vertical convolution with v_i for $1 \le i \le K$ is performed to obtain an output result y_{ln} for $1 \le l \le L - K + 1$ and $1 \le n \le N - K + 1$. Whereas each pixel is processed independently in the conventional two-dimensional convolutions as shown in Eq. (2), the previous separable convolution [10] employs separable property when a kernel w can be decomposed into vertical v and horizontal h kernels. In Figure 2, the structure for the separable two-dimensional convolution is represented as for $K = 3$, which consists of $2K$ multipliers, $2(K - 1)$ adders, $2K$ flip-flops, and K buffers with size of L to obtain one output pixel in a single cycle. The left-handed K multipliers and $(K - 1)$ adders are used to compute a horizontal convolution, and the right-handed K multipliers and $(K - 1)$ adders are used to compute vertical convolutions.

3. PROPOSED AREA-EFFICIENT STRUCTURE

In this section, we propose a new area-efficient structure by rescheduling the computational sequences. Although the previous separable structure [10] is succeeded in reducing the hardware resources remarkably by employing the separable property, it is possible to further optimize the structure by rescheduling the computations. First, let us analyze the vertical convolution of Eq. (2) in more details. When the size of kernel K is set to 3, Eq. (2) can be expanded as Eq. (3)

$$y_{ln} = v_1 t_{(l+1)n} + v_2 t_{(l+2)n} + v_3 t_{(l+3)n}. \tag{3}$$

As for the previous separable structure [10] of Fig. 2, the right-handed three multipliers are responsible for multiplying each temporary value with vertical convolution weight. However, most of those multipliers can be reduced by computing the multiplication at appropriate time if the temporary values are not generated simultaneously.

Thus, we propose a new area-efficient convolution structure by computing a vertical multiplication as soon as possible leading to the reduction of vertical multipliers. To describe the main algorithm of the proposed structure, Eq. (4) is expressed by inserting Eq. (2) into Eq. (3).

$$y_{ln} = \sum_{i=1}^{K} v_i \times \left(\sum_{j=1}^{K} h_i \times x_{(l+i)(n+j)} \right). \tag{4}$$

Whereas the previous separable convolution in Eq. (3) computes all the vertical multipliers at the same time, the proposed convolution in Eq. (4) computes one vertical multiplier at one time in a sequential manner. According to

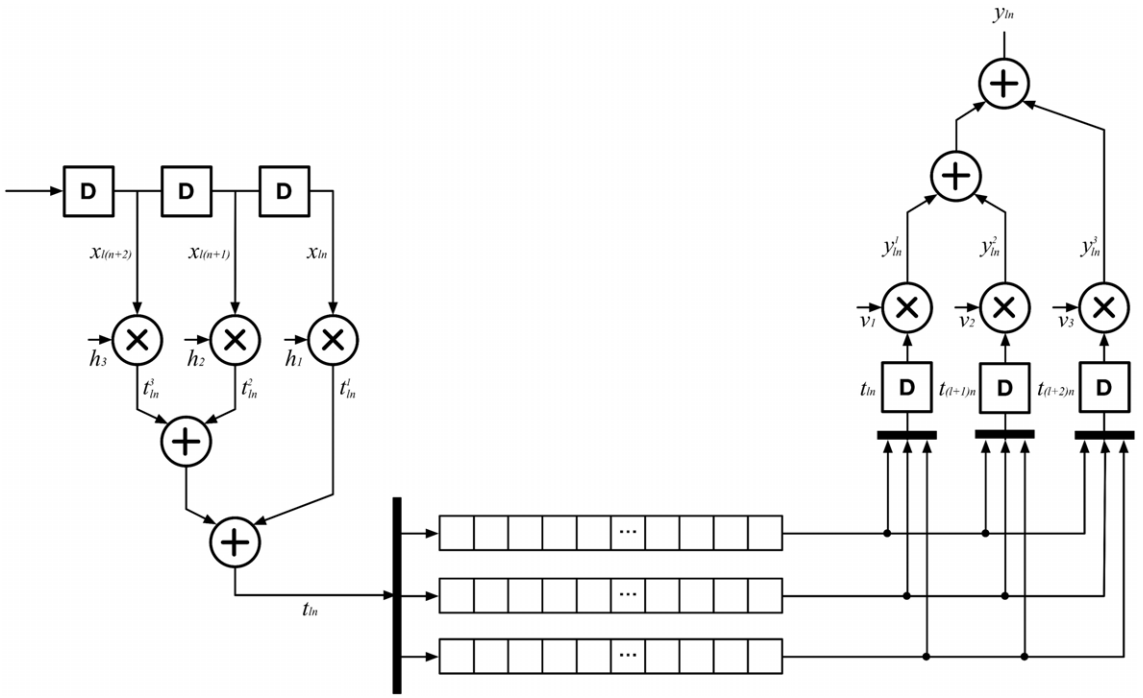

Figure 2. Structure of the separable two-dimensional convolution for $K = 3$.

Table I. Comparison hardware complexity with synthesis results.

Structure	Conventional [5] $\langle O(K^2) \rangle$			Separable [10] $\langle O(2K) \rangle$			Proposed $\langle O(K) \rangle$		
Kernel size	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7
Multiplier	5.0 k	14.0 k	27.4 k	2.3 k	5.7 k	8.1 k	2.2 k	3.4 k	4.6 k
Adder	1.1 k	3.3 k	6.6 k	0.6 k	1.2 k	1.7 k	0.6 k	1.2 k	1.7 k
Flip-flop	1.5 k	4.0 k	7.7 k	0.9 k	1.5 k	2.2 k	0.6 k	0.9 k	1.2 k
Total	7.6 k	21.3 k	41.7 k	3.8 k	8.4 k	12.0 k	3.4 k	5.5 k	7.5 k

Eq. (4), it is verified that one temporary value is generated at a time, thus the proposed structure can replace the multipliers of vertical convolutions with one multiplier that supports multiple weights. For example, three vertical multipliers with a fixed kernel weight can be replaced as a single vertical multiplier with versatile kernel weights.

Figure 3 describes the proposed structure for $K = 3$. The right-handed three multipliers and three multiplexers in the previous separable structure [10] of Fig. 2 are completely eliminated, and additional one multiplier and one mux is placed before the intermediate buffer in the proposed structure of Fig. 3. Note that an optional delay before the vertical multiplier can be inserted so as to prevent from lengthening the critical path as shown in Fig. 3.

4. EXPERIMENTAL RESULTS

To verify the advantages of the proposed structure, we have implemented various convolution architectures with different kernel sizes and compared with the conventional convolution structure [5] and the separable convolution structure [10]. For a fair comparison, all the convolution structures are synthesized in 65 nm CMOS technology with 200 MHz clock frequency, and Table I provides synthesis results in terms of equivalent gate counts. According to Table I, the proposed structure reduces the redundant usage of 80% and 38% of multipliers compared to the conventional [5] and previous separable structures [10]. Since the multipliers are the most dominant element in a convolution structure, the proposed structure always leads to the smallest hardware complexity. Furthermore, the hardware complexity in terms of the kernel size of K is summarized to investigate how efficiently the proposed method optimizes the hardware resources. According to Table I, the proposed structure always outperforms the conventional and the previous separable structure [10] for various kernel sizes based on big-O notation.

5. CONCLUSION

A new proposed convolution structure has been proposed to mitigate the complex hardware complexity caused by the increase of kernel and image sizes. The proposed structure firstly analyzes the previous separable convolution [10] and

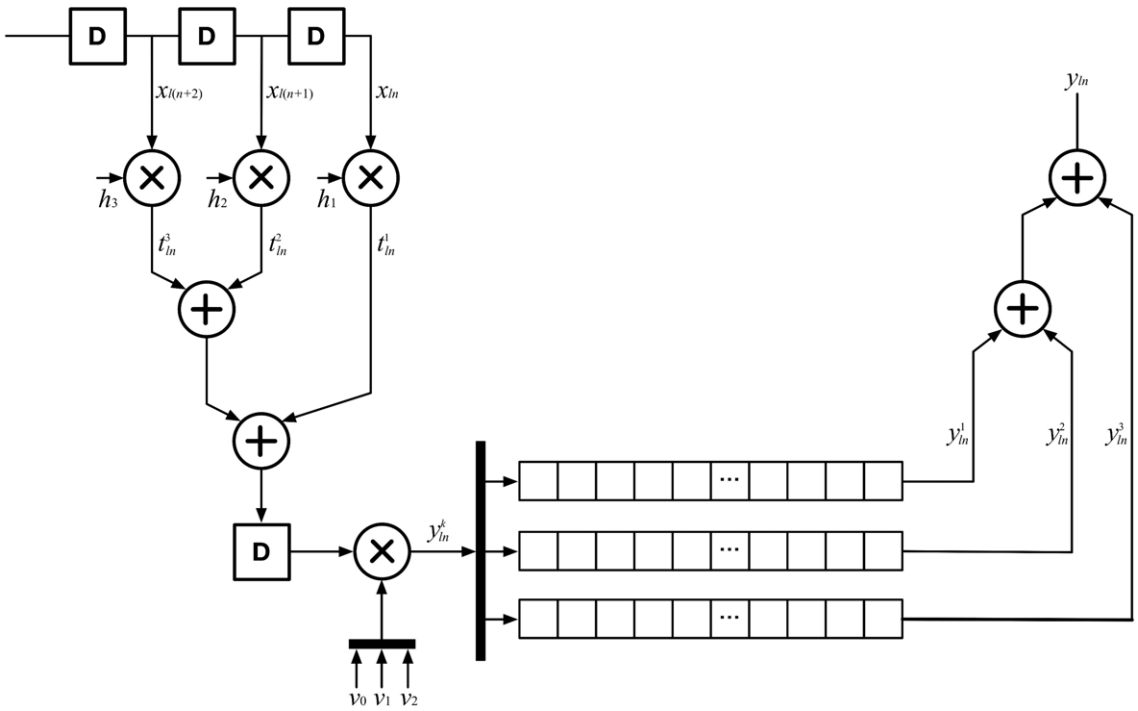

Figure 3. Structure of the proposed two-dimensional convolution for $K = 3$.

rescheduled the multiplications of the vertical convolution to save several multipliers. Experimental results show that the proposed structure guarantees the smallest hardware complexity without any performance degradation, and the improvement become more significant as the kernel size increases. Thus, the proposed structure can provide a solution to optimize severe convolutional hardware complexity.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation (NRF) grant funded by the Korea government (MSIP) (2017R1C1B5015962) and by the IC Design Education Center (IDEC).

REFERENCES

[1] A. Krizhevsky and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Neural Inf. Syst. Conf. **25** (2012).
[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. (Prientice Hall, Englewood Cliffs, NJ, 2002).
[3] J.-M. Yang and S.-H. Wei, "Spatial and spectral nonparametric linear feature extraction method for hyperspectral image classification," Adv. Technol. Innov. **2**, 68–72 (2016).
[4] R. Mehra and R. Verma, "Area efficient FPGA implemnetation of Sobel edge dectector for image processing applications," Int. J. Comput. Appl. **56**, 16 (2012).
[5] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Teman, "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *Proc. 19th Int'l. Conf. Archit. Support Program. Lang. Oper. Syst.* (ACM, New York, NY, 2014), pp. 269–284.
[6] J. Jo and I.-C. Park, "DSIP: A scalable inference accelerator for convolutional neural networks," IEEE Trans. J. Solid-State Circuits **53**, 605–618 (2018).
[7] J. Jo, S. Kim, and I.-C. Park, "Energy-efficient convolution architecture based on rescheduled dataflow," IEEE Trans. Circuits Syst. I **65**, 4196–4207 (2018).
[8] M. Kalbasi and H. Nikmehr, "A fine-grained pipelined 2-D convolver for high-performance application," IEEE Trans. Circuits Syst. II **66** (2019).
[9] J. Yue, Y. Liu, Z. Yuan, Z. Wang, Q. Guo, J. Li, C. Yang, and H. Yang, "A 3.77TOPS/W convolutional neural network processor with priority-driven kernel optimization," IEEE Trans. Circuits and Syst. II **66** (2019).
[10] D. Mukherjee and S. Mukhopadhyay, "Fast hardware architecture for 2D separable convolution operations," IEEE Trans. Circuits Syst. II **65**, 2042–2046 (2018).