Bin-Specific Quantization in Spectral-Domain Convolutional Neural Network Accelerators

Jinho Park*, Jaewon Lee†, Gain Kim†, and Hyeon-Min Bae*
*School of Electrical Engineering, KAIST, Daejeon 34141, Korea. {hg766, hmbae}@kaist.ac.kr
†Department of Electrical Engineering and Computer Science, DGIST, Daegu 42988, Korea. {ljw3136, gain.kim}@dgist.ac.kr

***Abstract*—Spectral-domain convolution engines can effectively reduce the computational complexity of convolution operations. In these engines, however, element-wise multiplications of the spectral representations dominate the multiply and accumulate (MAC) operations. In light of this, we propose bin-specific quantization (BSQ), which is to judiciously allocate varying bit width to each spectral bin in overlap-save. This allows efficient computation of the Hadamard product since the magnitude of the high-frequency components in image features is significantly smaller than that of the low-frequency counterparts. Using the statistics from spectral representations of feature maps, we also delineate methods for properly allocating bit precision to those spectral bins. When BSQ is applied, the average bit precisions of the arithmetic operators in spectral-domain convolvers, without the requirement of network re-training, were lowered by 24% (AlexNet), 20% (VGG–16), and 22% (ResNet–18) while having no significant reduction (<1%) on classification accuracy on the ImageNet dataset.**

***Index Terms*—Acceleration; Approximate Computing; Convolutional Neural Networks; Quantization; Retrain-less**

I. Introduction

While convolutional neural networks (CNN) showed unprecedented performance in machine learning problems such as image classification, object detection and natural language processing, data-driven nature of these schemes requires an excessive computation cost. Not only the sheer amount of the multiply and accumulate (MAC) but also the memory wall encountered when evicting data from the off-chip memory poses a severe bottleneck for real-time network inference at the edge. CNN inference accelerated on application-specific integrated circuits (ASICs) is several orders of magnitude more efficient than that of existing computing platforms on single-batch inference. This is achieved by dataflow optimization [1] and approximate computing which leverages inherent error resilience of NNs [2].

Although dot product-based convolvers are referred to as the *de facto* standard for CNN accelerators, another mode of convolving a feature map with a kernel by adopting the Fast Fourier Transform (FFT) demonstrates its benefits in terms of computational complexity over spatial-domain convolvers [3], [4]. Utilizing this approach reduces the number of MACs by 55% in case of VGG-16 (Fig. 1). Applying quantization further improves the throughput [5] while data reuse scheme to mitigate costly data movements for spectral convolvers is reported in [6].

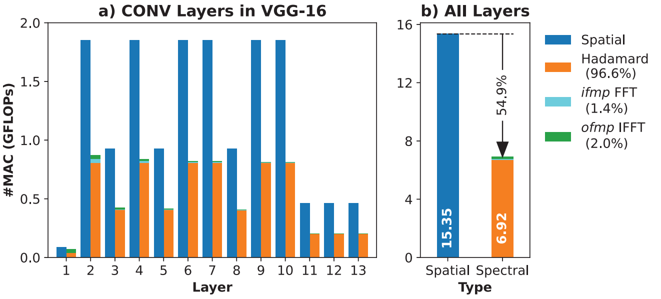


Fig. 1: (a) Computational complexity breakdown among all CONV layers of VGG-16. (b) Compares the total number of MACs for spatial- and spectral-domain convolutions, consisting of *ifmp* FFT, Hadamard product, and *ofmp* inverse FFT.

Nevertheless, reducing computation cost of Hadamard product is desirable as it accounts for 97% of all MAC operations involved in convolutional (CONV) layers of VGG-16 (Fig. 1). Because most of the energy in spectral representations of input (*ifmp*) and output (*ofmp*) feature maps is accommodated in low frequency bins, upper bits of high frequency bins are predominately zeros as illustrated in the following sections. Motivated by this, we leverage approximate computing wherein we truncate the upper bits and solely compute the lower bits, thereby saving the computation cost.

In this paper, we propose a novel method that effectively compresses the spectral representation of feature maps with quantization. The contributions of this work are threefold:

- We propose *Bin-Specific Quantization* (BSQ), which efficiently distributes varying bit precisions depending on the frequency bin. To the best of our knowledge, none of the prior arts exploits this property in overlap-save based neural networks.
- Methods for generating bin masks based on statistics of different neural networks are presented.
- The experimental results corresponding to optimization of spectral-domain quantization for CNNs such as AlexNet, VGG-16, and ResNet-18 on the ImageNet dataset are discussed in detail.

The rest of this paper is organized as follows. Section II gives a brief overview on works related to neural network deployment while Section III introduces spectral-domain CNNs. BSQ and methods for optimizing spectral-domain bin

978-1-6654-0996-4/22/$31.00 ©2022 IEEE

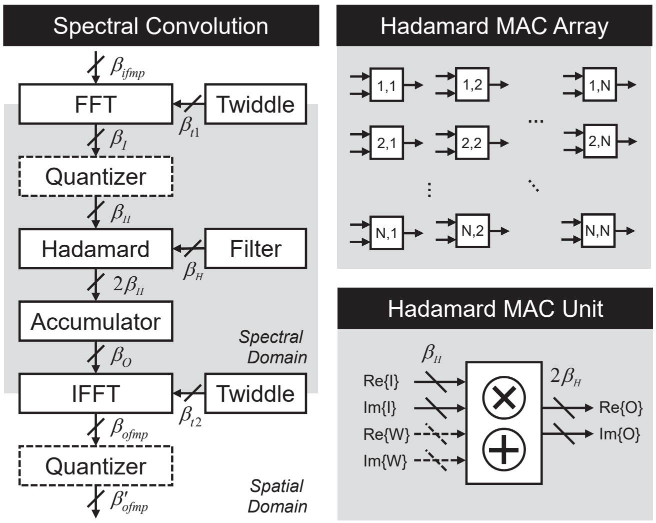


Fig. 2: Conceptual schematic to show dataflow in spectral-domain convolution wherein the Hadamard Array consisting of $N \times N$ MAC units.

masks are presented in Section IV, and the corresponding experimental results are discussed in Section V. Lastly, Section VI concludes the paper.

II. Related Works

For edge deployment of CNNs, lightweight NNs including MobileNet and ShuffleNet are developed. These works mainly focus on bringing down the number of model parameters and MACs by replacing standard convolutions with depth-wise separable or group convolutions. Despite the success of lightweight NNs in image classification, 2D CONV-based NNs such as VGG and ResNet still serve as the backbones of NNs for object detection and image segmentation.

CNN inference can also be accelerated with approximate computing techniques. First of all, quantization mitigates the memory access overhead by reducing the size of feature maps and kernels [7]. This lowers memory bandwidth burden [2] as well as the computation cost with a negligible drop in classification accuracy. Next, efficient algorithms for convolutions are also explored in prior arts. Since element-wise multiplication in spectral-domain is the dual of convolution in spatial-domain, spectral CNN accelerator utilizing this duality have been proposed [3], [4]. Sun *et al.* [5] adopts spectral-domain algorithm in tandem with appropriate quantization technique [7] to further lighten the cost of multiplication. To combat the kernel explosion issue, a dataflow optimized for spectral convolvers is reported in [6].

III. Spectral-Domain Convolution

In this section, we briefly describe the procedures for spectral-domain convolution based on overlap-save, shown in Fig. 2. After being divided into tiles of size $N \times N$, a scale-quantized *ifmp* is transformed to spectral-domain via N-tap 2D FFT. The processing gain from FFT is managed by a quantizer, truncating the bit-precision to be β_H-bit. By letting $\beta_{ifmp} = \beta'_{ofmp} = 9$-bit in our simulation, the Rectified Linear Unit (ReLU) forces our feature maps to be unsigned 8-bit in the spatial-domain. Also, if $N = 16$, $\beta_H = \beta_{ifmp} + 2\log_2 N = 17$-bit due to the 2D FFT processing gain. Prepared in the same manner as *ifmp*, the kernel (filter) can be pre-calculated to streamline real-time inference. Spectral *ifmp*s (I) and spectral kernels (W) of size $N \times N$ are element-wise multiplied in the Hadamard MAC Array (Fig. 2). Notice how each MAC Unit, which is a complex multiplier, takes and produces real and imaginary components for each input and output with coupled bit precisions (β_H or $2\beta_H$). These partial products are accumulated along the input channel dimension. The resulting *ofmp* in spectral-domain is converted back into spatial-domain by 2D IFFT followed by discarding and abutting according to OS. Detailed procedure for spectral-domain convolution involving overlap-add is explained in detail in [5].

Despite the scheduling overhead needed for domain conversion, spectral convolution reduces the total number of computations (Fig. 1). By simplifying the derivation in [4], computational complexity for each type of convolutions in a given layer can be written as:

$$\Omega_{\text{Spatial}} = H^2 \cdot R^2 \cdot C_{in} \cdot C_{out}$$

$$\Omega_{\text{Spectral}} = N^2 \cdot T \cdot \big(\underbrace{c_1 \cdot \log N \cdot C_{in}}_{\textit{ifmp}\text{ FFT}} + \underbrace{c_2 \cdot C_{in} \cdot C_{out}}_{\text{Hadamard}} + \underbrace{c_3 \cdot \log N \cdot C_{out}}_{\textit{ofmp}\text{ IFFT}}\big)$$

where H is the feature map size, R denotes the kernel size, $C_{in,out}$ correspond to input and output channel numbers, and $T = \left\lceil \frac{H}{N-R+1} \right\rceil^2$ is the number of tiles. As the layer proceeds, $C_{in}, C_{out} >> \log N$, dominating the computation workload as seen in Fig. 1. We choose 16-tap FFT ($N = 16$) as this strikes a good balance between the number of computation per feature and memory access.

IV. Bin-Specific Quantization

In this section, we explain the rationale behind BSQ and then delineate the methods for BSQ. We first illustrate how spectral representations of *ifmp* have more energy in bins closer to the DC. Quantized, spectral representations of *ifmp* (I) for all CONV layers in AlexNet, VGG-16, and ResNet-18 are attained and shown in Fig. 3. Here, an $N \times N$ tile is flattened out along the x-axis. Samples across channel, tile, and batch dimensions are aggregated to generalize the distribution. For example, in the first layer of VGG-16, $(T, C_{in}) = (256, 3)$ from $(H, N, R) = (224, 16, 3)$. On the other hand, since $R = 7$, $T = 529$ in the first layer of ResNet-18. As a result, each spectral bin in $N \times N$ tile has total of 0.55M, 4.63M, and 1.13M samples from 100-image subset of the ImageNet for AlexNet, VGG-16, and ResNet-18, respectively.

In Fig. 3, the distribution of each network manifests its unique profile. More importantly, the smallest 100-percentile (maximum value) among spectral bins in AlexNet is 9.57-bit, implying upper bits in the high frequency bins are unoccupied. It can also be seen that 10-bit (signed 11-bit) is enough to accommodate 75% of the distributions (Q_3) for all network

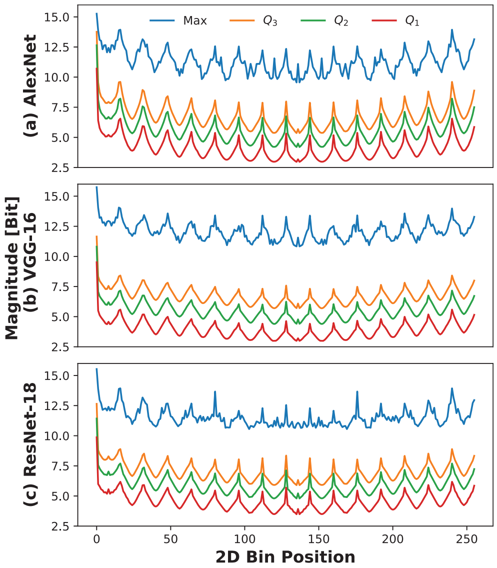
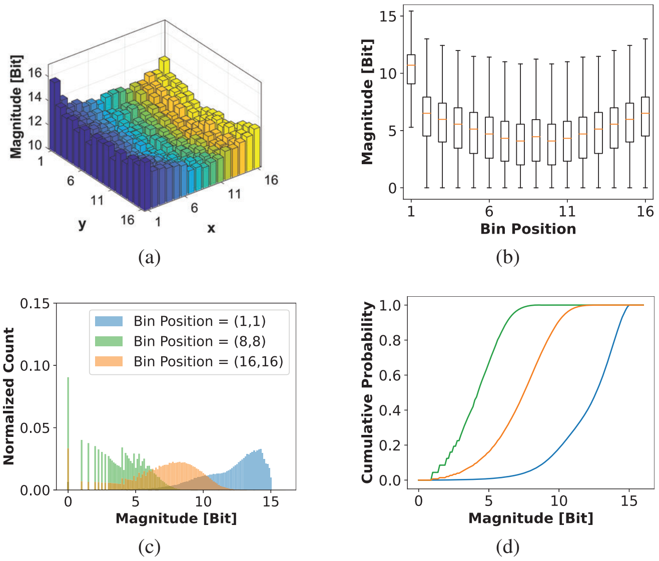


Fig. 3: The interquartile range and the maximum value of $\log_2 |\mathrm{Re}\{I(x,y)\}|$ at flattened 2D bin positions across all layers of (a) AlexNet, (b) VGG-16, and (c) ResNet-18.



(a)

(b)

(c)

(d)

Fig. 4: The spectral bins of *ifmp*s collected from all CONV layers of VGG-16 shown as (a) 3D contour of the maximum values of each bin, (b) box plot of the diagonal entries e.g. $x = y \in \{1, 2, ..., 16\}$, (c) histograms at specified bin positions, and (d) their corresponding cumulative distribution functions.

types, except for the DC bin. Fig. 4 illustrates bin-specific distributions for VGG-16. Both 2D contour of maximum values of spectral bins (Fig. 4a) and box-plot of diagonal entries from spectral bins (Fig. 4b) are analogous representations of Fig. 3b. The energy contained in higher frequency components is considerably smaller compared to that of the lower frequency components as in Fig. 4(c,d).

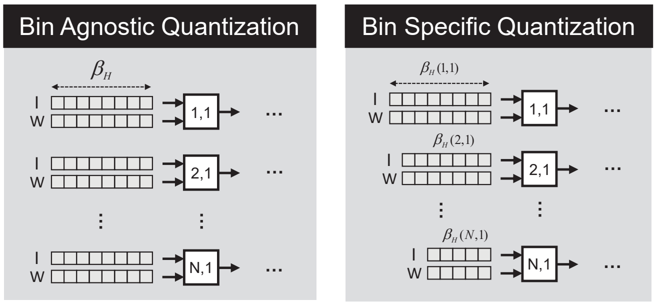


Fig. 5: Hadamard MAC Array has an homogeneous bit precision in Bin Agnostic Quantization (left), but varying bit precisions in Bin Specific Quantization (right).

To that end, BSQ harnesses above-mentioned quantization opportunities by approximating the computations of the high frequency components, shown in Fig. 5. Unlike the conventional approach (Bin Agnostic Quantization), BSQ truncates upper bits in high frequency bins, which are otherwise wasted. Although BSQ does not improve the throughput given the same number of MAC cores, it does help in reducing the area of each core in ASIC implementation, resulting in power saving as well. Just like any other quantization schemes, the extent to which the bit precisions can be lowered without accuracy loss must be evaluated.

In order to demonstrate the effect of BSQ, we present methods for generating bin masks for Hadamard product computation. The same 2D bit precision map would be used throughout the inference time, regardless of image, layer, channel, and tile. We carefully surmise that the distributions of each bin implies the extent to which its bit width can be truncated. For this, $|\mathrm{Re}\{I\}|$ has been chosen as the reference statistics since the real part is bigger than the imaginary part, and spectral bins of *ifmp* are greater than that of the kernel. Three different methods for generating bin masks $\beta_H(x,y)$ are as follows:

1) MAX – It takes the maximum value of the bin profile, which can be visualized as Fig. 4a.
2) CDF – The bit-precision of each bin is mapped according to the given cumulative probability (Fig. 4d).
3) BF – Similar to Greedy Algorithms, the brute force approach lowers the bit precision $\beta_H(x,y)$ in each bin one by one until the performance metric reaches below the threshold classification accuracy.

This aforementioned framework is implemented in Python by applying the parameters of the pre-trained models from PyTorch without network retraining. Such software-based framework is crucial as it complements hardwired implementation by giving preliminary performance evaluation.

V. Experimental Results

The effect of BSQ with different bin masks on the accuracy and bit-precision trade-off is reported in this section. After evaluating the performance on a smaller dataset from which we have gathered bin-specific statistics, we rerun the test

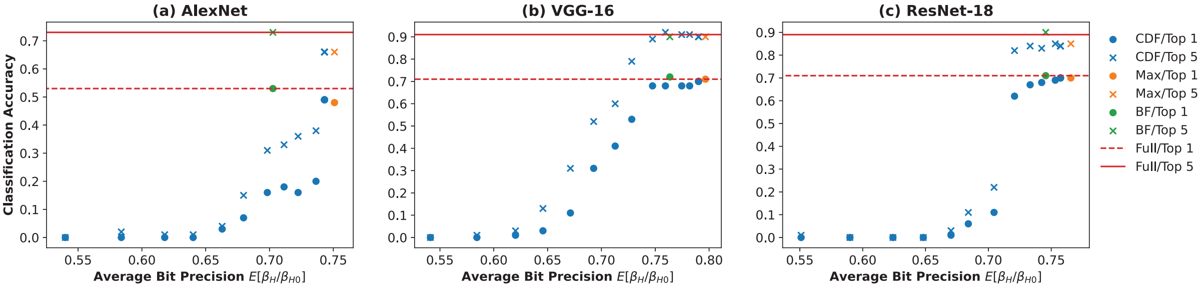


Fig. 6: The impact of bit precision β_H on image classification accuracy of (a) AlexNet, (b) VGG-16, and (c) ResNet-18.

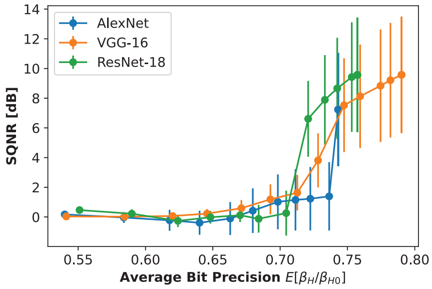


Fig. 7: The effect of bit precision β_H on *ofmp* SQNR.

TABLE I: Effect of bin mask on classification accuracy

Model	Bin Mask	$\mathbb{E}[\beta_H/\beta_{H0}]$	Accuracy Drop	
			Top-1	Top-5
AlexNet	BF	0.703	2.58%	1.26%
	MAX	0.751	15.4%	9.02%
	BF+MAX	0.759	0.87%	-0.01%
	CDF	0.743	16.82%	9.81%
VGG-16	BF	0.764	13.73%	7.31%
	MAX	0.797	1.52%	0.71%
	BF+MAX	0.801	0.13%	0.02%
	CDF	0.759	6.94%	3.49%
ResNet-18	BF	0.746	5.12%	3.30%
	MAX	0.766	5.05%	3.15%
	BF+MAX	0.776	0.59%	0.29%
	CDF	0.757	5.71%	3.26%

with a much larger set. Image classification accuracies for a subset of 100 images sampled from the validation set of the ImageNet with different masking schemes are shown in Fig. 6. Discontinuities in classification accuracies with respect to the average bit precision are evident when the expected value of bit-ratio $\mathbb{E}[\beta_H/\beta_{H0}] < 0.74$ for AlexNet and $\mathbb{E}[\beta_H/\beta_{H0}] < 0.72$ for ResNet-18. Here, a 2D bin mask of varying bit-precisions $\beta_H(x, y)$ of shape 16×16 are averaged and then divided by baseline fixed-point bit precision $\beta_{H0} = 17$-bit. Unlike those two, accuracy drop-off is more gradual in case of VGG-16 when $\mathbb{E}[\beta_H/\beta_{H0}] < 0.75$. Such accuracy roll-off can also be seen in the signal to quantization noise ratio (SQNR) of *ofmp* as illustrated in Fig. 7.

The same procedure has been repeated on the ImageNet subset with 10k images, and the results are summarized in Table I. Though the statistics were generated from a smaller subset, because each spectral bin contained a large amount of samples (e.g. 4.63M for VGG-16), it is generalizable to a much larger set. If accuracy drop can be accepted to a degree, bins generated with BF and CDF methods can achieve larger bit-savings. If not, the bin mask generated by combining BF and MAX, i.e. $\max(\beta_{\text{BF}}, \beta_{\text{MAX}})$, yields Top-1/5 classification accuracy drop less than 1% compared to the full bit quantization scheme without BSQ. In this case, average bit precisions amongst 256 bins were reduced by 24.13% (AlexNet), 19.88% (VGG-16), and 22.40% (ResNet-18). Since narrowing down the bit precision quadratically decreases the area of a MAC unit [3] when implemented as ASICs, energy- and area-saving for the Hadamard MAC Array can be expected.

VI. Conclusion

In this work, we propose methods for quantizing spectral bins in the Hadamard MAC Array, which takes the dominant workload in a spectral-domain accelerator. The bin-specific quantization appropriately decides bin-specific bit widths based on the bin statistics. Towards this, we have characterized the spectral bin distributions of intermediate feature maps across all layers of different networks. With statistics-based bin generation, BSQ achieves average bit precision reduction of 24%, 20%, and 22% for AlexNet, VGG-16, and ResNet-18, respectively.

References

[1] Y.-H. Chen *et al.*, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[2] J. Lee *et al.*, "Link bit-error-rate requirement analysis for deep neural network accelerators," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[3] C. Zhang *et al.*, "Frequency domain acceleration of convolutional neural networks on cpu-fpga shared memory system," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017, pp. 35–44.

[4] H. Zeng *et al.*, "A framework for generating high throughput cnn implementations on fpgas," in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2018, pp. 117–126.

[5] W. Sun *et al.*, "Throughput-optimized frequency domain cnn with fixed-point quantization on fpga," in *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. IEEE, 2018, pp. 1–8.

[6] Y. Niu *et al.*, "Reuse kernels or activations? a flexible dataflow for low-latency spectral cnn acceleration," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2020, pp. 266–276.

[7] D. Lin *et al.*, "Fixed point quantization of deep convolutional networks," in *International conference on machine learning*. PMLR, 2016, pp. 2849–2858.